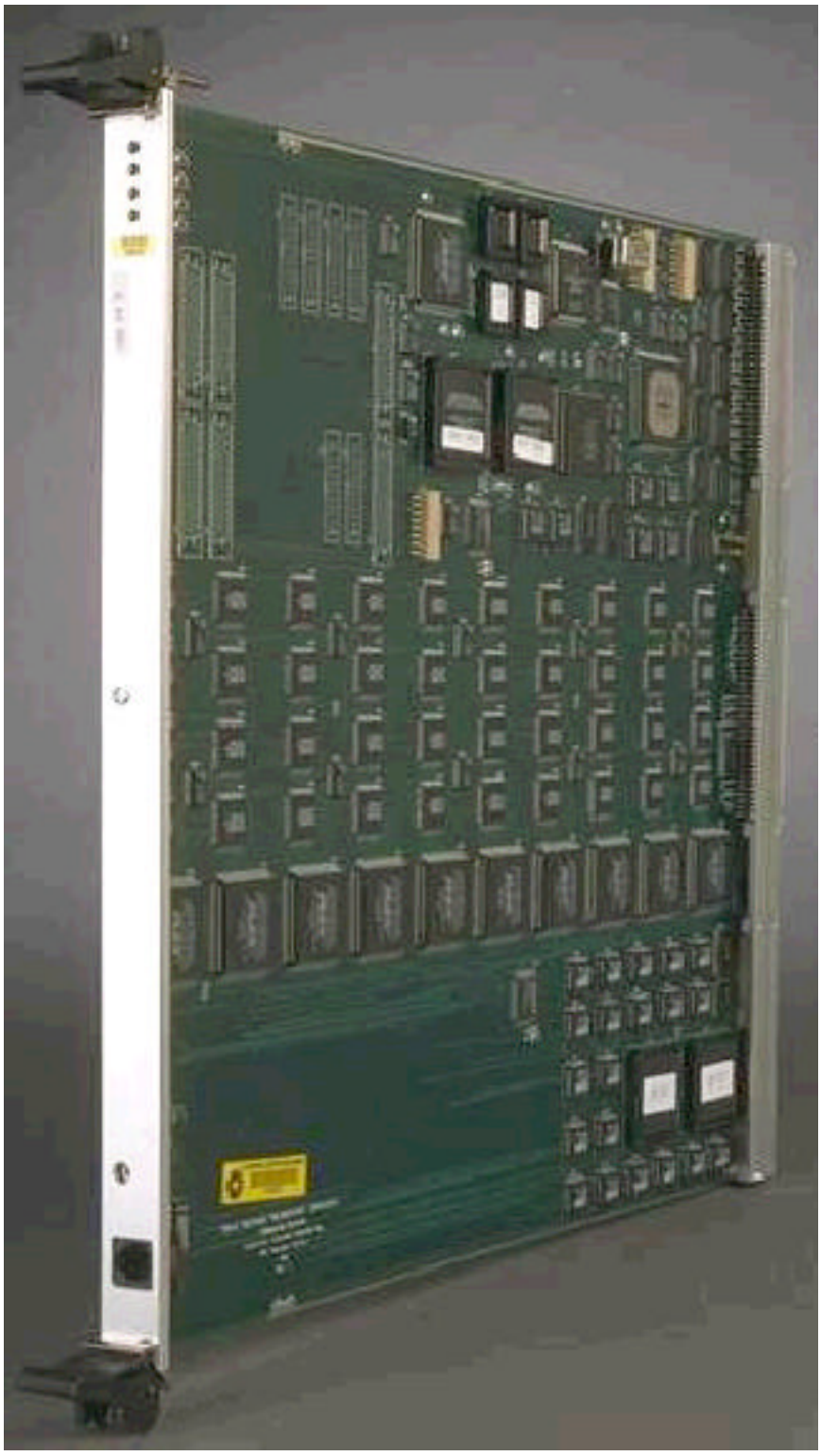
VME READOUT BUFFER (VRB)

Date: 10/12/2001

Document # ESE-SVX-950719



VRB

VRB Updates

(Listed in reverse chronological order. Intermediate versions may have been generated for test purposes.)

Date	File Name	Application	Changes
10/06/01	Vrb1B06P.hex	0 (D0 SVX)	- no changes
		1 (CDF SVX)	- 20 usec timeout on data arrival - delete leading zero words in event stream - assert error line 3 for timeout - add "wysiwyg" mode (no zero word deletion, no EOR deletion, no "D0" deletion) - add enable register for "wysiwyg" mode - add enable register for timeout
		2 (CDF DAQ)	- no changes
		3 (CDF TEST)	- no changes
		4 (D0 TEST)	- no changes
		5 (D0 Trigger)	- no changes
7/25/01	Vrb1725P.hex	0 (D0 SVX)	- Move non-pipelined block data starting address to 0xC00000, (increase pipelined block address range to 32Kbytes)
		1 (CDF SVX)	- no changes
		2 (CDF DAQ)	- no changes
		3 (CDF TEST)	- no changes
		4 (D0 TEST)	- no changes
		5 (D0 Trigger)	- no changes
2/08/01	vrb1208P.hex	0 (D0 SVX)	- add control bit to disable backplane status signals when board is not in use - change default buffer configuration to 16 X 2K (buffers 0-15) with overlapping 8 X 4K set (buffers 16-23) - enable software write-protect

		1 (CDF SVX)	- enable software write-protect
		2 (CDF DAQ)	- enable software write-protect
		3 (CDF TEST)	- enable software write-protect
		4 (D0 TEST)	- enable software write-protect
		5 (D0 Trigger)	- add control bit to disable backplane status signals when board is not in use - change default buffer configuration to 16 X 2K (buffers 0-15) with overlapping 8 X 4K set (buffers 16-23) - enable software write-protect
11/12/00	vrb0B12.hex	0 (D0 SVX)	- no changes
		1 (CDF SVX)	- no changes
		2 (CDF DAQ)	- restore original BUSY code and fix problem which could cause missing EOR word under certain BUSY conditions, also add input FIFO AFF to BUSY
		3 (CDF TEST)	- no changes
		4 (D0 TEST)	- no changes
		5 (D0 Trigger)	- no changes
10/25/00	vrb0A25.hex	0 (D0 SVX)	- fix for missing first words in D64 block transfer
		1 (CDF SVX)	- fix channel order in header status words to match new channel ordering
		2 (CDF DAQ)	- no changes
		3 (CDF TEST)	- fix channel order in header status words to match new channel ordering
		4 (D0 TEST)	- revert to original channel ordering to avoid hangups on unpopulated channels
		5 (D0 Trigger)	- fix for missing first words in D64 block transfer
10/19/00	vrb0A19.hex	0 (D0 SVX)	- remove previous VME changes due to possible FIFO conflict

		1 (CDF SVX)	- remove previous VME changes due to possible FIFO conflict
		2 (CDF DAQ)	- add hysteresis to BUSY signal, remove internal VRB holds on BUSY - remove previous VME changes due to possible FIFO conflict
		3 (CDF TEST)	- remove previous VME changes due to possible FIFO conflict
		4 (D0 TEST)	- remove previous VME changes due to possible FIFO conflict
		5 (D0 Trigger)	- remove previous VME changes due to possible FIFO conflict
10/05/00	vrb0A05.hex	0 (D0 SVX)	- modify scan-ready status signal so that it is asserted as soon as the output FIFO begins to fill (instead of waiting for the complete event to be transferred) - changes to VME interface to improve response time, reduce bus errors
		1 (CDF SVX)	- changes to VME interface to improve response time, reduce bus errors - eliminate BIT3 specific VME logic
		2 (CDF DAQ)	- changes to VME interface to improve response time, reduce bus errors - eliminate BIT3 specific VME logic
		3 (CDF TEST)	- changes to VME interface to improve response time, reduce bus errors - eliminate BIT3 specific VME logic
		4 (D0 TEST)	- changes to VME interface to improve response time, reduce bus errors - eliminate BIT3 specific VME logic
		5 (D0 Trigger)	- modify scan-ready status signal so that it is asserted as soon as the output FIFO begins to fill (instead of waiting for the complete event to be transferred) - changes to VME interface to improve response time, reduce bus errors
9/01/00	vrb0901.hex	0 (D0 SVX)	no changes
		1 (CDF SVX)	- fix to prevent misread of SRC command during VME read of status registers
		2 (CDF DAQ)	no changes
		3 (CDF TEST)	- fix to prevent misread of SRC command during VME read of status registers

		4 (D0 TEST)	- fix to prevent misread of SRC command during VME read of status registers
		5 (D0 Trigger)	no changes
8/28/00	vrb0828.hex	0 (D0 SVX)	- add monitor functions (same functionality as cdf svx monitor) - set bit 0 of third header word if gray decoding enabled. Remove bit per channel gray decode enable and replace with single bit (bit 0) in enable register - add "User Info" (16 bits) to upper half of second header word - add module ID (geographical address) to top byte of lower half of second header word - add firmware version number to upper half of third header word - expand EOR character to "C0" (instead of "Cx"), to avoid conflict with sequencer ID - disable gray code decoding for all incoming bytes which are not SVX chip data. In addition, chip data is not decoded when the chip ID is 14 or 15 ("virtual" SVX chip) - removed code which prevented restart mode change
		1 (CDF SVX)	no changes
		2 (CDF DAQ)	no changes
		3 (CDF TEST)	no changes
		4 (D0 TEST)	no changes
		5 (D0 Trigger)	- set bit 1 of third header word to indicate trigger record - remove automatic EOR when input FIFO empty > 10 clocks - add "User Info" (16 bits) to upper half of second header word - add module ID (geographical address) to top byte of lower half of second header word - add firmware version number to upper half of third header word

7/07/00	vrb0707.hex	0 (D0 SVX)	no changes
		1 (CDF SVX)	no changes
		2 (CDF DAQ)	no changes
		3 (CDF TEST)	no changes
		4 (D0 TEST)	no changes
		5 (D0 Trigger)	- added D0 Trigger application (application 5). Initially the same as D0 SVX mode, except using G-Link D17/D19 instead of "Cx" for EOR
6/06/00	vrb0606.hex	0 (D0 SVX)	no changes
		1 (CDF SVX)	- reversed order of check for scan commands, so that VRB expects scan buffer number before event number
		2 (CDF DAQ)	no changes
		3 (CDF TEST)	no changes
		4 (D0 TEST)	no changes
4/03/00	vrb0403.hex	0 (D0 SVX)	- remove check of output FIFO empty which allowed VME D64 access to FIFO before data ready (similar to previous problem, but different cause)
		1 (CDF SVX)	- remove check of output FIFO empty which allowed VME D64 access to FIFO before data ready (similar to previous problem, but different cause)
		2 (CDF DAQ)	- remove check of output FIFO empty which allowed VME D64 access to FIFO before data ready (similar to previous problem, but different cause)
		3 (CDF TEST)	- remove check of output FIFO empty which allowed VME D64 access to FIFO before data ready (similar to previous problem, but different cause)

		4 (D0 TEST)	- remove check of output FIFO empty which allowed VME D64 access to FIFO before data ready (similar to previous problem, but different cause)
3/17/00	vrb0317.hex	0 (D0 SVX)	- modify arbitration to avoid VME D64 read of output FIFO before data is ready
		1 (CDF SVX)	- modify arbitration to avoid VME D64 read of output FIFO before data is ready - set controller error bit (status signal 6) if VRB receives a read buffer number command without first receiving both pipeline capacitor number and bunch crossing number commands, or if VRB receives a scan buffer number command without first receiving an event number command. (this is to better detect corrupted commands on J3 backplane) - fixed error which caused status information for channels 1 and 6 not to appear in output header - fixed "time since level 1" in header to reflect value at readout command rather than last value received. ("time since level 1" was being overwritten if additional read commands were received before scan)
		2 (CDF DAQ)	- modify arbitration to avoid VME D64 read of output FIFO before data is ready
		3 (CDF TEST)	- modify arbitration to avoid VME D64 read of output FIFO before data is ready
		4 (D0 TEST)	- modify arbitration to avoid VME D64 read of output FIFO before data is ready

1/07/00 (vrb0107.hex) Production baseline version.

General Description

The VRB is a multiport memory designed to buffer and filter data for transfer to online processors. It contains ten independent input ports and a common VME output port. The VRB receives data via application specific transition module data links, which are typically serial optical connections. Buffer management may be provided by the VRB internal logic or by an external System Controller through either VME or a dedicated control port.

The VRB can accept input data at an aggregate rate of approximately 500 MBytes/sec on ten (byte-wide) channels. The output rate is limited by VME transfer speeds and by the number of VRB modules sharing the VME bus. To make optimum use of the module, a significant trigger rejection factor between input and output event rates is assumed. The VRB buffer memory may be partitioned into fixed length buffers of arbitrary size or may be programmed to act in a FIFO mode.

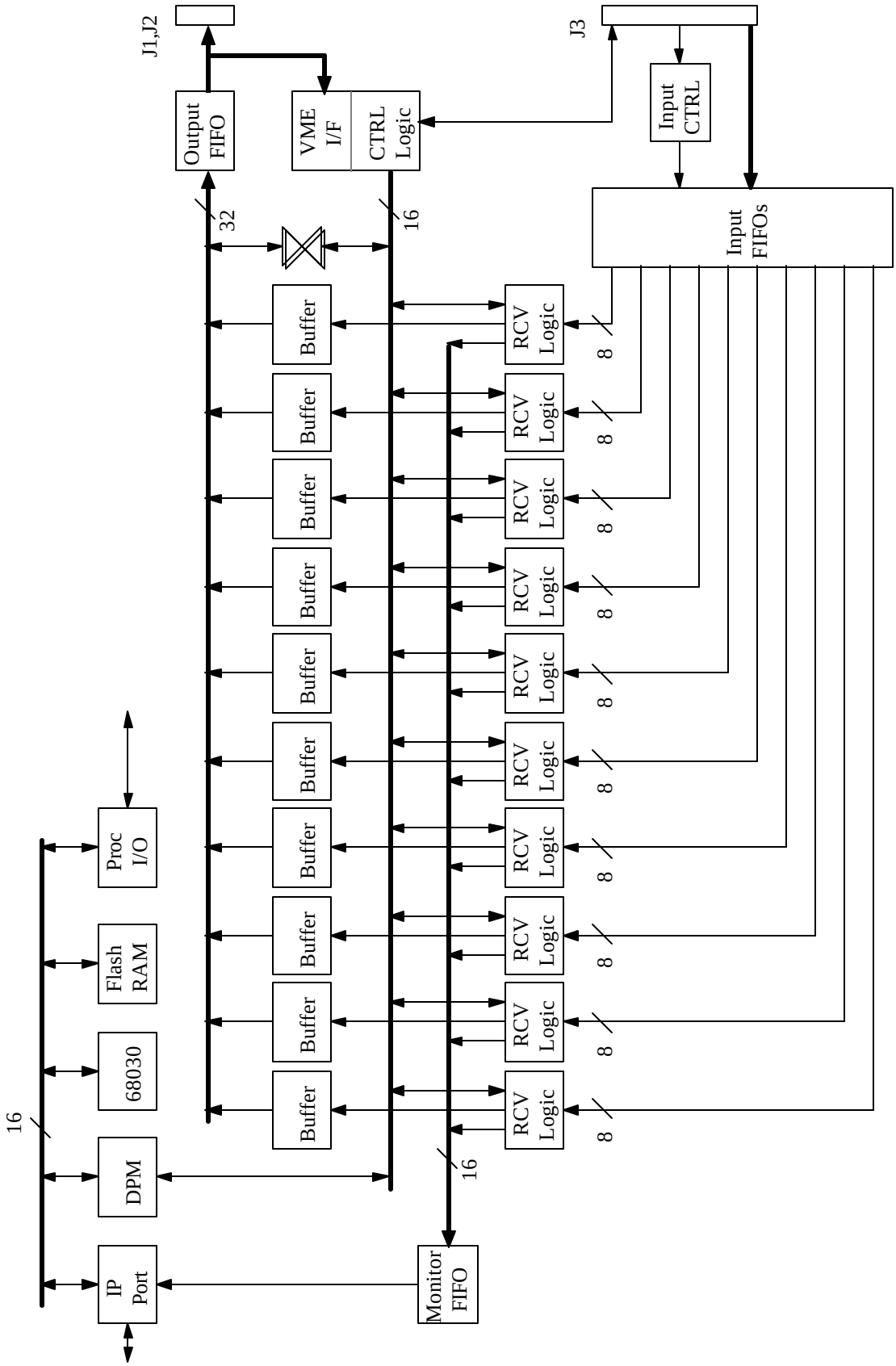
Design requirements for the VRB module are based on readout rates specified for the DO and CDF detector upgrades. For SVX applications, the VRB must input data at the L1 Accept rate (50 KHz), and must output data at the L2 Accept rate (~1 KHz). For some applications, the VRB must input data at the L2 Accept rate and output data at the same rate. Event size varies with the application, from ~300 Bytes per channel for SVX to several KBytes per channel for other applications.

Features

- 9U x 400mm "VME" board.
- Independent external control port (no VME control messages required for buffer management).
- 8 or 10 data channels
- Input data rate - 400-500 MBytes/sec (50 MBytes/sec/channel).
- Output data rate - 50 MBytes/sec
- Programmable buffer sizes and buffer operating modes.
- Dual-port operation (simultaneous read and write of event data buffers).

Each VRB receives up to 10 streams of event data from the front-end system and buffers that data for VME readout to the Level 3 trigger system. The VME connection is shared by all VRB modules in a subrack. For a subrack with 10 VRB modules and 50% input link occupancy, there is an effective VRB input to output bandwidth ratio of 50:1. This approximates the expected SVX L1/L2 accept ratio. For other applications, slower input links or fewer VRBs per subrack may be necessary to properly balance the dataflow.

A block diagram of the VRB is shown in the following figure:


J1,J2
J3
Output FIFO
VME I/F
CTRL Logic
Input CTRL
Input FIFOs
32
16
Buffer
RCV Logic
8
Proc I/O
Flash RAM
68030
DPM
IP Port
Monitor FIFO
16

VRB Functional Description

Individual functional blocks are described in this section.

Control Logic

The VRB Control Logic performs three basic functions; receive and process messages from the System Controller or VME port, return status signals to the System Controller, and manage the general flow of data to and from the VRB buffers

Messages from the System Controller are received on the P5/6 connector and buffered in a small FIFO where they are processed in the order received. Messages consist of one or more bytes, each byte accompanied by a 4 bit identifier. The messages are application dependent. The Control Logic will respond to a message by asserting a status signal, typically within 200 nsec.

When the VRB Control Logic receives a message specifying the next input buffer, it looks up the buffer starting address in the shared memory, and broadcasts this information to the Receive Logic for all channels. When all event data is received, the VRB Control Logic will read the individual byte counts from each Receive Logic block and generate a global byte count for the event. The global byte count is available to a VME Scan Processor so that it can perform a single block read operation to obtain all data for the event. During output operations, the individual channel byte counts are used by the VRB Control Logic to initialize a DMA controller that concatenates and transfers data blocks to the VME output FIFO.

VME Logic

The VME slave interface logic and VRB Control Logic are contained in the same FPGA. The VME logic is entirely programmable and can support a variety of existing or future VME modes within the limitation of available FPGA resources. VME operation is application dependent (see Application section).

Processor

The VRB uses a Motorola 68EC030 processor to initialize on-board programmable logic and to set the default operating parameters. The processor is not involved directly in the data transfer operations.

The VRB processor has access to 1 MByte of non-volatile (Flash) memory, arranged in two banks;

Bank 0 contains the VRB "operating system" (default processor startup code and default programmable logic configurations to allow communication with the VRB through VME).

Bank 1 contains processor code and programmable logic configurations for all of the individual applications.

Both banks may be write protected by on-board switches.

At startup the VRB will examine the setting of the Application Select 3 switch (S3-8). If this switch is OFF, the VRB will be initialized with default code to allow generic VME access and diagnostics. If the switch is ON, the processor will examine the other Application Select switches and branch to the corresponding application entry point in Bank 1. The VRB is then initialized according to the requirements of the specific application.

Shared Memory

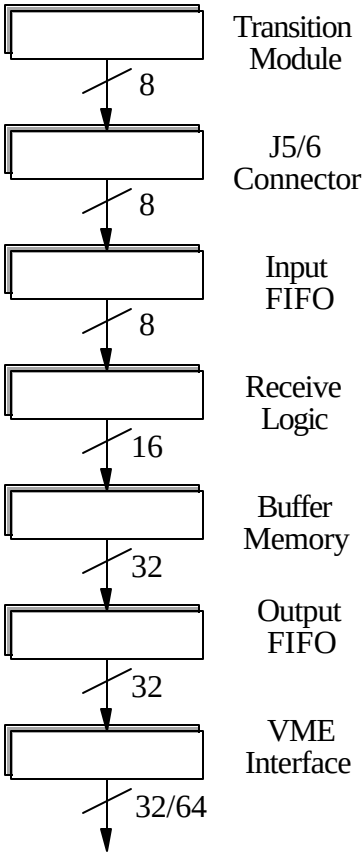
A 16 KByte shared memory (DPM) allows communication between the Control Logic and Processor. During initialization the processor copies operating parameters (starting addresses, etc) from non-volatile memory to shared memory for use by the Control Logic. During processing, the Control Logic uses the shared memory to store detailed status information and byte counts for each event. The shared memory is also used by the processor for local temporary data storage. Access to this memory from VME is arbitrated by the Control Logic.

Input Data Ports

The VRB supports 8 or 10 input channels, each channel receiving a stream of bytewide parallel data. In addition to the 8 data bits, each channel receives 2 control bits which can be used for start or end-of-record identifiers, etc. This control information is not stored with the data.

Input FIFOs are used to decouple the link/VRB data streams. The input FIFOs will hold up to 512 bytes per VRB channel. This allows for some skew between the arrival of the first byte of data and receipt of the control message specifying the VRB buffer for data input. If the event data exceeds 512 bytes per channel, the control message must be received by the VRB within approximately 6-7 microseconds (512 bytes/53 MBytes sec minus setup overhead) of the start of data transmission or the input FIFO may overflow.

The following figure illustrates the datapath for a single VRB channel. The data is received as a bytewide stream and then expanded to 16 bits at the input of the Receive Logic. Data is stored in buffer memory in 16 bit increments (with padding to 64 bit boundaries). The buffer memory output is a common 32 bit bus for all ten channels. Output data is held temporarily in a 32 bit output FIFO and then transferred to VME in either 32 bit or 64 bit transactions.



Receive Logic

Event data passes through the Receive Logic before it is stored in the VRB buffers. The Receive Logic program is application dependent. It performs event format verification and can also be used for limited data processing (e.g., Gray code conversion).

Event Buffers

The VRB supports a programmable number of event buffers (up to 64). Buffer starting locations and sizes are also programmable and are aligned on 8 Byte boundaries. The SARAM memory specified for the VRB provides a total of either 32 or 64 KBytes per input channel.

The buffer numbers to be used for input and output may be identified by the System Controller or generated internally by the VRB. To accelerate VME output operations, the VRB calculates a total byte count for each of the stored events. A VME master determines how much data to transfer by reading the total byte count, and then performing a block read from the output data FIFO port. Data from all enabled VRB buffers is automatically concatenated into one VME block.

Output Data FIFO

The VRB includes a 16 KByte output data FIFO to decouple the event buffer memory from VME. Events may be pipelined in this output FIFO, allowing release of the SCAN BUSY signal in synchronized applications as soon as the event has been copied to the FIFO (without waiting for completion of the VME transfer).

Monitor FIFO

The VRB provides a 4 K X 16 bit Monitor FIFO which allows sampling of event data from a single VRB input channel (for diagnostics or statistical evaluation). The information can then be sent to the VRB Auxiliary Port (or to VME accessible buffers). The Auxiliary Port consists of a control FPGA and mezzanine connectors for a dual width "Industry Pack" or user-defined mezzanine card. In the D0 SVX application, this mezzanine card will likely hold a 1394 serial interface. In other applications it may be used as an Ethernet interface. Monitor FIFO data may be transferred to the Auxiliary Port at a relatively high rate (using DMA logic in the FPGA). Normal reads and writes to this port from the processor will be slower.

A monitor event is selected by writing a VRB channel number to the Monitor Channel register. The Control Logic will poll this information when initializing the Receive Logic at the start of each event readout. At the completion of event readout, the Control Logic copies the byte counts for all channels to the shared memory, and sets the Monitor Channel register to a specified value to indicate that the requested event data is available. The Monitor FIFO may be enabled at any time, but does not begin recording data until the start of the next event readout.

The monitoring system interface is intended to provide an auxiliary control and diagnostics port which is independent of the VME bus. This minimizes interference with VME data transfers for systems that support this interface. All communication with the VRB through this port is handled by the VRB processor.

System Controller Interface

A System Controller is not required for VRB operation, but it will improve performance in most applications by removing control traffic from the VME bus during data acquisition. It also allows a centralized interface to the trigger system so that event data can be read from the VRB in the correct order without knowing the VRB buffer number associated with each event.

The pinout for the System Controller Interface on the VRB P5/6 connector is provided in the following table (pins designated “tm” are feedthroughs used by the transition module input data links);

PIN	ROW A	ROW B	ROW C	ROW D	ROW E
1	tm	tm	msg0	tm	tm
2	tm	tm	msg1	tm	tm
3	tm	tm	msg2	tm	tm

4	tm	tm	msg3	tm	tm
5	tm	tm	msg4	tm	tm
6	tm	tm	msg5	tm	tm
7	tm	tm	msg6	tm	tm
8	tm	tm	msg7	tm	tm
9	tm	tm	msg8	tm	tm
10	tm	tm	msg9	tm	tm
11	tm	tm	msg10	tm	tm
12	tm	tm	msg11	tm	tm
13	tm	tm	msg_strobe	tm	tm
14	tm	tm	status0	tm	tm
15	tm	tm	status1	tm	tm
16	tm	tm	status2	tm	tm
17	tm	tm	status3	tm	tm
18	tm	tm	status4	tm	tm
19	tm	tm	status5	tm	tm
20	tm	tm	status6	tm	tm
21	tm	tm	status7	tm	tm
22	tm	tm	status8	tm	tm
23	tm	tm	status9	tm	tm
24	tm	tm	reserved	tm	tm
25	tm	tm	reserved	tm	tm
26	tm	tm	GND	tm	tm
27	tm	tm	reserved	tm	tm
28	tm	tm	reserved	tm	tm
29	tm	tm	GND	tm	tm
30	tm	tm	reserved	tm	tm
31	tm	tm	reserved	tm	tm
32	tm	tm	GND	tm	tm
33	tm	tm		tm	tm
34	tm	tm		tm	tm
35	tm	tm		tm	tm
36	tm	tm	tm	tm	tm
37	tm	tm	tm	tm	tm
38	tm	tm	tm	tm	tm
39	tm	tm	tm	tm	tm
40	tm	tm	tm	tm	tm
41	tm	tm		tm	tm
42	tm	tm		tm	tm
43	tm	tm		tm	tm
44	tm	tm		tm	tm

45	tm	tm	tm	tm	tm
46	tm	tm	tm	tm	tm
47	tm	tm	tm	tm	tm

The MSG and STATUS bus are AC terminated. Pullups for the open collector STATUS bus are on the System Controller module.

The message data (MSG[11:0]) is written into the VRB message FIFO on the rising edge of the MSG_STROBE signal. Message data should be valid for at least 15 ns (setup time) before the strobe.

Program Memory

The Flash memory is not directly accessible through VME, but may be updated by the on-board processor in response to requests from VME.

The flash memory is arranged in two banks. The first bank (bank 0) contains startup code which is generic to all applications. The second bank (bank 1) contains up to 8 applications. Each bank can be individually write protected.

Processor

The 68EC030 code is written in C. FPGA data files are linked into the C code as const segments.

FPGAs

The Control Logic, Receive Logic and Auxiliary Port FPGAs are Altera 8000 series components. The Processor control, Processor I/O port and Input Port control PLDs are Altera 7064s. All devices are programmed using the Altera MAX PLUS+ development system. The PLD/FPGA code is written in AHDL, compiled and the resulting Altera ".ttf" files are then merged with the 60EC030 code.

The following device types are used for programmable logic;

Control Logic	EPF81188A
Receive Logic	EPF8820A
Auxiliary Port	EPF8820A
Processor Control	EPM7064

Processor I/O	EPM7064
Input Port Control	EPM7064

Diagnostic Interface

A simple 9600 Baud, RS232 compatible terminal port is present on the front panel of the VRB. This port is driven directly by the processor, using a software emulation of a UART. It is used to display configuration information along with any startup error conditions detectable by the processor. Information will be displayed only if the Diagnostic Port Disable switch (S3-3) is OFF.

Panel Indicators

The VRB has four front panel LED indicators.

Power LED:

ON	Normal operation
OFF	No module power

Processor LED:

ON	Normal operation
OFF	VRB has failed to initialize
Blinking	*Application dependent*

VME Access LED:

ON	VME access

Controller Access LED:

ON	System Controller access
Blinking	*Application dependent*

VRB Clock Generator

The VRB is normally clocked by a 50 MHz local oscillator. Most of the on-board logic operates at 25 MHz. There is a software option to drive all VRB logic, other than the processor, from a processor generated clock (single-step).

Auxiliary Port

TBD

Test Ports

The VRB printed circuit board layout provides a number of individual diagnostic test points distributed across the module. These are identified in the following illustration:

TBD

There are also six groups of test points arranged for simple connection to HP logic analyzer probes. The following signals are available in these groups:

Probe Chan #	**J16**	**J17**	**J18**	**J19**	**J20**	**J21**
D0	PA16	GND	PD16	RA1	CD0	GND
D1	PA17	PA1	PD17	RA2	CD1	CA1
D2	PA18	PA2	PD18	RA3	CD2	CA2
D3	PA19	PA3	PD19	RA4	CD3	CA3
D4	PA20	PA4	PD20	RA5	CD4	CA4
D5	PA21	PA5	PD21	PAS*	CD5	CA5
D6	PA22	PA6	PD22	PDS*	CD6	CA6
D7	PA23	PA7	PD23	PRD	CD7	CA7
D8	PA24	PA8	PD24	PDSACK*	CD8	CA8
D9	PA25	PA9	PD25	PInt 1*	CD9	CA9
D10	PA26	PA10	PD26	PInt 2*	CD10	CA10
D11	PA27	PA11	PD27	DPM RD	CD11	CA11
D12	Flash 0*	PA12	PD28	Monitor*	CD12	CA12
D13	DPM*	PA13	PD29	IPA ack	CD13	CA13
D14	Aux port*	PA14	PD30	CRD	CD14	n/c
D15	I/O port*	PA15	PD31	25 MHz	CD15	n/c

CLK1	25 MHz	25 MHz	25 MHz	25 MHz	n/c	25 MHz

Pxx = Processor Bus

Rxx = Receive Logic Address Bus

Cxx = Control Bus

Development Software

Applications required for software/firmware development:

Altera MAX PLUS+ II development system

Crossware Products 680x0 C Compiler (requires a parallel port key)

Altera Development:

All Altera code is in AHDL and is located in the directory "vrb_fpga". There is a separate subdirectory for each PLD/FPGA and application version. Altera ".tdf" files are compiled by the MAX PLUS+ II software to generate ".ttf" object files. The latest versions of the ".ttf" files for all Altera devices are automatically combined with the VRB processor embedded program (via #include statements and C const data arrays) whenever the VRB C source code is recompiled.

Embedded C Code:

C code development uses the Crossware Products embedded C compiler. The VRB C source code is in the directory "vrb_sw".

VRB Programming using External Programmer

The output of the compiler is an S-record file ("vrb.hex") which may be used to program the flash memory via the Logical Devices AllPro or similar programmer.

To program the low bank of flash memory using the AllPro;

1) turn the AllPro programmer Off and On,

   wait for it to boot (it must be rebooted each time it is used)

2) double click the AllPro application on the PC
3) "boot from host" by pressing the "UP" button on the AllPro programmer,
   then press the "Enter" button on the AllPro programmer,
   wait for the host boot to complete
4) from the "Device" Menu, choose "Select /F6"
5) set the "Manufacturer" to ATMEL,
   set the "Type" to EEPROM,
   set the "Package" to PLCC,
   click the down arrow in the device select list, scroll down and select "AT29C020"
   click OK
6) in the "Device Memory options:" window,
   set "Devices in Set:" to 2, click OK
7) click the "DnLoad" button in the main window
8) in the "File DnLoad" window, choose the "Motorola Hex" file format,
   locate the vrb.hex file, select the file and click OK
9) in the "PROM Options" window,
   in the "File Address" section, set the "Word size" to 16 and click OK,
   wait ~10 minutes for file download
10) insert the first flash memory chip as shown on the programmer
11) if necessary, click on the "<" button under "Data Set: 1/2" to select set 1 of 2
12) click on the "Prog." button in the main window, wait for programming to complete
13) remove the first chip and insert the second flash memory chip
14) click on the ">" button under "Data Set: 1/2" to select set 2 of 2
15) click on the "Prog." Button in the main window,
    wait for programming to complete
16) remove the second chip

The high bank of flash memory may also be programmed using the AllPro programmer. All steps are identical to those listed above, except in step 9,in the "File Address" section, set "Start:" to 1000000.

The chips are placed on the VRB with the first chip on the left and the second chip on the right for each bank. Bank 0 is the lower set of flash memory chips. Bank 1 is the upper set.

Transition Modules

The VRB uses transition modules for the input data interface and System Controller interface.

The input data link transition module allows use of application specific link protocols. Transition modules with G-Link and TAXI optical receivers are currently available.

The System Controller transition module allows operation of multiple VRB subracks from a single controller. Alternatively, a separate System Controller can reside in each VRB subrack or the VRB can be operated without an external controller.

VRB Fanout (System Controller Transition Module)

The VRB Fanout (VFO) is a front mounted transition module which connects a System Controller in the same VRB subrack or in a different subrack to the VRB modules. This connection is made using front panel ribbon cable between the System Controller and the VFO. The VFO then repeats the signals on a special J5/6 backplane in the VRB subracks.

The VFO resides in slot 14 of a VRB subrack and translates low voltage differential signals (LVDS) on the cable to TTL signals on the J5/6 backplane. There are two groups of signals; a control bus which is used by the System Controller to send commands to the VRBs, and a status bus used by the VRBs to signal completion of event processing and various error conditions.

The VFO is designed to allow use of a single System Controller with multiple VRB subracks. In applications where a separate controller is used in each VRB subrack, the VRB Fanout Module is not required.

The pinout for the VRB Fanout J5/6 (slot 14) connection is provided in the following table.

PIN NUMBER	ROW A	ROW B	ROW C	ROW D	ROW E
1			msg0		
2			msg1		
3			msg2		
4			msg3		
5			msg4		
6			msg5		

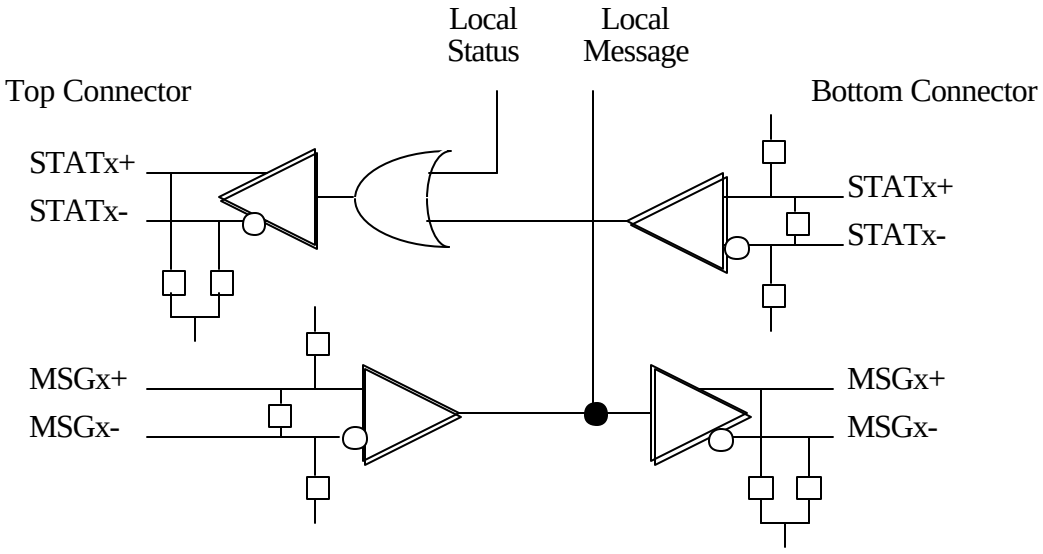
7			msg6		
8			msg7		
9			msg8		
10			msg9		
11			msg10		
12			msg11		
13			msg_strobe		
14			status0		
15			status1		
16			status2		
17			status3		
18			status4		
19			status5		
20			status6		
21			status7		
22			status8		
23			status9		
24			reserved		
25			reserved		
26			GND		
27			GND		
28			GND		
29			GND		
30			GND		
31			GND		
32			+5V		
33					
34					
35					
36					
37					
38					
39					
40					
41					
42					
43					
44					
45					
46					
47					

The following signals appear on the front-panel connectors as LVDS levels. Pin assignments for the front-panel connector are listed in the following table. Both ribbon cable connectors use the same pin assignment. All signals are point-to-point and are terminated on the receiving end at each Fanout module or System Controller. A ribbon cable connects the System Controller to the top connector of the first VRB Fanout module. Each additional VRB Fanout module is linked with a separate ribbon cable from the bottom connector of the previous VRB Fanout module to the top connector of the next VRB Fanout. Status signals are inverted in the VRB Fanout (low true on the backplane, high true on the cable).

PIN NUMBER	**SIGNAL**
1	MSG0-
2	MSG0+
3	MSG1-
4	MSG1+
5	MSG2-
6	MSG2+
7	MSG3-
8	MSG3+
9	MSG4-
10	MSG4+
11	MSG5-
12	MSG5+
13	MSG6-
14	MSG6+
15	MSG7-
16	MSG7+
17	MSG8-
18	MSG8+
19	MSG9-
20	MSG9+
21	MSG10-
22	MSG10+
23	MSG11-
24	MSG11+
25	STROBE-
26	STROBE+
27	STAT0-
28	STAT0+

29	STAT1-
30	STAT1+
31	STAT2-
32	STAT2+
33	STAT3-
34	STAT3+
35	STAT4-
36	STAT4+
37	STAT5-
38	STAT5+
39	STAT6-
40	STAT6+
41	STAT7-
42	STAT7+
43	STAT8-
44	STAT8+
45	STAT9-
46	STAT9+
47	GND
48	GND
49	GND
50	GND

The Fanout repeater circuit is shown in the following figure. All signals are point-to-point, terminated LVDS.



The VRB Fanout also provides a set of HP logic analyzer compatible connectors for monitoring the state of VME bus signals. The module does not otherwise connect to or use VME.

G-Link Transition Module

The G-Link transition module holds four Finisar optical receivers and four HP G-Link decoders which may operate in either 16 or 20 bit modes.

In 16 bit mode, two data streams are encoded on each G-Link fiber to provide a total of eight bytewide virtual links. In this mode, two of the ten VRB data channels are not used. No control inputs are available (control information is embedded in the data stream). This is the D0 SVX default mode.

In 20 bit mode, two and a half data streams are encoded on each G-Link fiber to provide a total of ten bytewide virtual links. All ten VRB data channels are used. No control inputs are available (control information is embedded in the data stream). This is the CDF SVX default mode.

The G-Link transition module may also be operated in 20 bit mode to provide eight bytewide data channels, each with 2 bits of control input for applications where the control information cannot be embedded in the data stream. In this mode, two of the ten VRB channels would again not be used. This is the D0 Trigger default mode.

Each virtual link operates at approximately 53 MBytes/sec in the SVX applications. Virtual links sharing the same physical G-Link fiber operate independently, but there is some interaction which must be considered in the Receive Logic.

The P5/6 connector pin definitions for a G-Link transition module are listed in the following table (pins designated “sc” are reserved for System Controller signals on the backplane).

PIN	ROW A	ROW B	ROW C	ROW D	ROW E
1	L0_D0	GND	sc	GND	L2_D0
2	L0_D1	L0_D2	sc	L2_D2	L2_D1
3	GND	L0_D3	sc	L2_D3	GND
4	L0_D5	L0_D4	sc	L2_D4	L2_D5
5	L0_D6	GND	sc	GND	L2_D6
6	L0_D7	L0_D8	sc	L2_D8	L2_D7
7	GND	L0_D9	sc	L2_D9	GND
8	L0_D11	L0_D10	sc	L2_D10	L2_D11
9	L0_D12	GND	sc	GND	L2_D12
10	L0_D13	L0_D14	sc	L2_D14	L2_D13
11	GND	L0_D15	sc	L2_D15	GND
12	L0_D16	L0_D16	sc	L2_D16	L2_D16
13	L0_D17	GND	sc	GND	L2_D17

14	L0_D17	L0_D18	sc	L2_D18	L2_D17
15	GND	L0_D18	sc	L2_D18	GND
16	L0_D19	L0_D19	sc	L2_D19	L2_D19
17	L0_CAV*	GND	sc	GND	L2_CAV*
18	L0_DAV*	L0_LNKRDY*	sc	L2_LNKRDY*	L2_DAV*
19	GND	L0_LNKRDY*	sc	L2_LNKRDY*	GND
20	L0_STRBOUT	GND	sc	GND	L2_STRBOUT
21	L0_SIG_DETECT	L0_ERROR	sc	L2_ERROR	L2_SIG_DETECT
22	GND	L0_ERROR	sc	L2_ERROR	GND
23	GND	GND	sc	GND	GND
24	GND	GND	sc	GND	GND
25	GND	GND	sc	GND	GND
26	L1_CAV*	GND	sc	GND	L3_CAV*
27	L1_DAV*	L1_LNKRDY*	sc	L3_LNKRDY*	L3_DAV*
28	GND	L1_LNKRDY*	sc	L3_LNKRDY*	GND
29	L1_STRBOUT	GND	sc	GND	L3_STRBOUT
30	L1_SIG_DETECT	L1_ERROR	sc	L3_ERROR	L3_SIG_DETECT
31	GND	L1_ERROR	sc	L3_ERROR	GND
32	L1_D0	GND	sc	GND	L3_D0
33	L1_D1	L1_D2		L3_D2	L3_D1
34	GND	L1_D3		L3_D3	GND
35	L1_D5	L1_D4		L3_D4	L3_D5
36	L1_D6	GND	GND	GND	L3_D6
37	L1_D7	L1_D8	GND	L3_D8	L3_D7
38	GND	L1_D9	GND	L3_D9	GND
39	L1_D11	L1_D10	GND	L3_D10	L3_D11
40	L1_D12	GND	GND	GND	L3_D12
41	L1_D13	L1_D14		L3_D14	L3_D13
42	GND	L1_D15		L3_D15	GND
43	L1_D16	L1_D16		L3_D16	L3_D16
44	L1_D17	GND		GND	L3_D17
45	L1_D17	L1_D18	SERIAL	L3_D18	L3_D17
46	GND	L1_D18	RESET*	L3_D18	GND
47	L1_D19	L1_D19	MODID	L3_D19	L3_D19

The MODID (module ID) pin is left unconnected to identify a G-Link transition module.

The data links are configured in simplex mode with a system supplied feedback path that informs the G-link transmitters of the synchronization state of the G-link receivers. If SYNC_ERROR is asserted by the VRB, the System Controller sends a message to the data sources requesting transmission of fill frames. The System Controller then waits until SYNC_ERROR is deasserted. In some cases it may be necessary to reset the VRB to force G-Link receiver synchronization.

Details of the VRB G-Link Transition module (VTM) design are found in a separate document.

TAXI Link Transition Module

The TAXI transition module holds ten AMP optical receivers and ten AMD TAXI decoders. In addition there are ten optical transmitters for returning "busy" signals to the data sources. The receivers normally operate in 10 bit mode. This provides ten bytewide connections, each with 2 bits of control information. This is the CDF DAQ default mode.

Each link operates at approximately 15 Mbytes/sec.

P5/6 connector pin definitions for a TAXI transition module are listed in the following table (pins designated "sc" are reserved for System Controller signals on the backplane.....a System Controller is currently used only in SVX applications);

PIN	ROW A	ROW B	ROW C	ROW D	ROW E
1	L0_D0	GND	sc	GND	L4_D0
2	L0_D1	L0_D2	sc	L4_D2	L4_D1
3	GND	L0_D3	sc	L4_D3	GND
4	L0_D5	L0_D4	sc	L4_D4	L4_D5
5	L0_D6	GND	sc	GND	L4_D6
6	L0_D7	L1_D0	sc	L5_D0	L4_D7
7	GND	L1_D1	sc	L5_D1	GND
8	L1_D3	L1_D2	sc	L5_D2	L5_D3
9	L1_D4	GND	sc	GND	L5_D4
10	L1_D5	L1_D6	sc	L5_D6	L5_D5
11	GND	L1_D7	sc	L5_D7	GND
12	L0_D8	L8_D0	sc	L9_D0	L4_D8
13	L0_D9	GND	sc	GND	L4_D9
14	L8_D1	L1_D8	sc	L5_D8	L9_D1
15	GND	L8_D2	sc	L9_D2	GND
16	L1_D9	L8_D3	sc	L9_D3	L5_D9
17	L0_DSTRB	GND	sc	GND	L4_DSTRB
18	L1_DSTRB	L0_BUSY	sc	L4_BUSY	L5_DSTRB
19	GND	L1_BUSY	sc	L5_BUSY	GND
20	L0_CLOCK	GND	sc	GND	L4_CLOCK
21	L1_CLOCK	L0_VLTN	sc	L4_VLTN	L5_CLOCK
22	GND	L1_VLTN	sc	L5_VLTN	GND
23	L8_CLOCK	GND	sc	GND	L9_CLOCK
24	L8_DSTRB	L8_VLTN	sc	L9_VLTN	L9_DSTRB

25	GND	L8_BUSY	sc	L9_BUSY	GND
26	L2_DSTRB	GND	sc	GND	L6_DSTRB
27	L3_DSTRB	L2_BUSY	sc	L6_BUSY	L7_DSTRB
28	GND	L3_BUSY	sc	L7_BUSY	GND
29	L2_CLOCK	GND	sc	GND	L6_CLOCK
30	L3_CLOCK	L2_VLTN	sc	L6_VLTN	L7_CLOCK
31	GND	L3_VLTN	sc	L7_VLTN	GND
32	L2_D0	GND	sc	GND	L6_D0
33	L2_D1	L2_D2		L6_D2	L6_D1
34	GND	L2_D3		L6_D3	GND
35	L2_D5	L2_D4		L6_D4	L6_D5
36	L2_D6	GND	L8_D8	GND	L6_D6
37	L2_D7	L3_D0	L8_D9	L7_D0	L6_D7
38	GND	L3_D1	GND	L7_D1	GND
39	L3_D3	L3_D2	L9_D8	L7_D2	L7_D3
40	L3_D4	GND	L9_D9	GND	L7_D4
41	L3_D5	L3_D6		L7_D6	L7_D5
42	GND	L3_D7		L7_D7	GND
43	L8_D4	L2_D8		L6_D8	L9_D4
44	L2_D9	GND		GND	L6_D9
45	L8_D5	L3_D8	(reserved)	L7_D8	L9_D5
46	GND	L8_D6	RESET*	L9_D6	GND
47	L3_D9	L8_D7	MODID	L9_D7	L7_D9

The MODID pin is pulled low to identify a TAXI transition module.

Details of the TAXI Transition module design are found in a separate document.

Electrical & Mechanical Specifications

Packaging

The VRB is a 9U x 400mm VME module. The printed circuit board is FR4 with a thickness of approximately .092". The top and bottom card edges are milled to .063". The board is 10 layers with nominal 50 ohm controlled impedance.

The VRB Fanout (VFO) is a 9U x 400mm module. It connects to the VME backplane for power only. The board is made of FR4 with a thickness of .092". Top and bottom card edges are milled to .063".

Power and Fusing

VRB power consumption varies from 15 watts (idle state) to approximately 40 watts (operating). The module requires only +5 volt DC power, which is supplied on 12 connector pins (3 in P1, 3 in P2 and 6 in P0). The pins are derated to 1.5A each, for a total current capacity of 18A. The module is fused at 10A (single picofuse). It also contains a General Semiconductor 5KP5.0A transient suppressor (5000 watt, 6.4 volt trip) on the load side of the fuse.

VTM power consumption is approximately 20 watts. This module requires only +5 volt DC power, which is supplied on 6 connector pins (all in P0). The pins are derated to 1.5A each, for a total current capacity of 9A. The module is fused at 5A (single picofuse). It also contains a 1N5908 transient suppressor (1500 watt, 6.2 volt trip) on the load side of the fuse.

VFO power consumption is approximately 10 watts. This module requires only +5 volt DC power, which is supplied on 3 connector pins (all in P1). The pins are derated to 1.5A each, for a total current capacity of 4.5A. The module is fused at 2A (single picofuse). It also contains a 1N5908 transient suppressor (1500 watt, 6.2 volt trip) on the load side of the fuse.

Application 0 (DO SVX)

In D0 SVX mode, the VRB acts as a buffer for data pending a Level 2 trigger decision. Buffer management is provided by a controller (the VRBC) which resides in slot 14 of each VME subrack. A subrack may contain up to 12 VRB modules. Each VRB receives data from four detector layers via the "SVX Sequencer" and G-Link "VTM".

The VRB startup application and some programming/diagnostic features are controlled by switch bank S3;

S3-1 and S3-2 are write protect switches for the VRB flash memory. These may normally be left OFF (not write protected) to enable in-system updates of the VRB firmware and software. Since the VRB software does not allow flash write cycles when in D0 SVX mode, there is generally no danger of accidentally overwriting the flash memory.

Switch S3-3 disables the RS-232 terminal diagnostic port. This port outputs information at VRB startup/reset and during some error conditions. If there is no terminal connected, this switch should be ON (disabled), since the VRB takes longer to recognize and execute a reset command when it is displaying diagnostic messages.

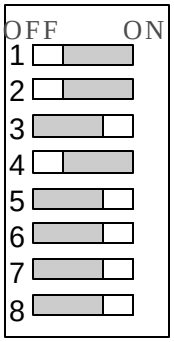
Switch S3-4 disables the remote mode change feature. When enabled (switch OFF), the VRB can be remotely rebooted into a mode other than D0 SVX. This is necessary for remote reprogramming.

Switches S3-5 through S3-8 are used to select the VRB startup application. Application select switch settings for D0 SVX mode are as follows;

	S3-8	**S3-7**	**S3-6**	**S3-5**
D0 SVX	ON	ON	ON	ON

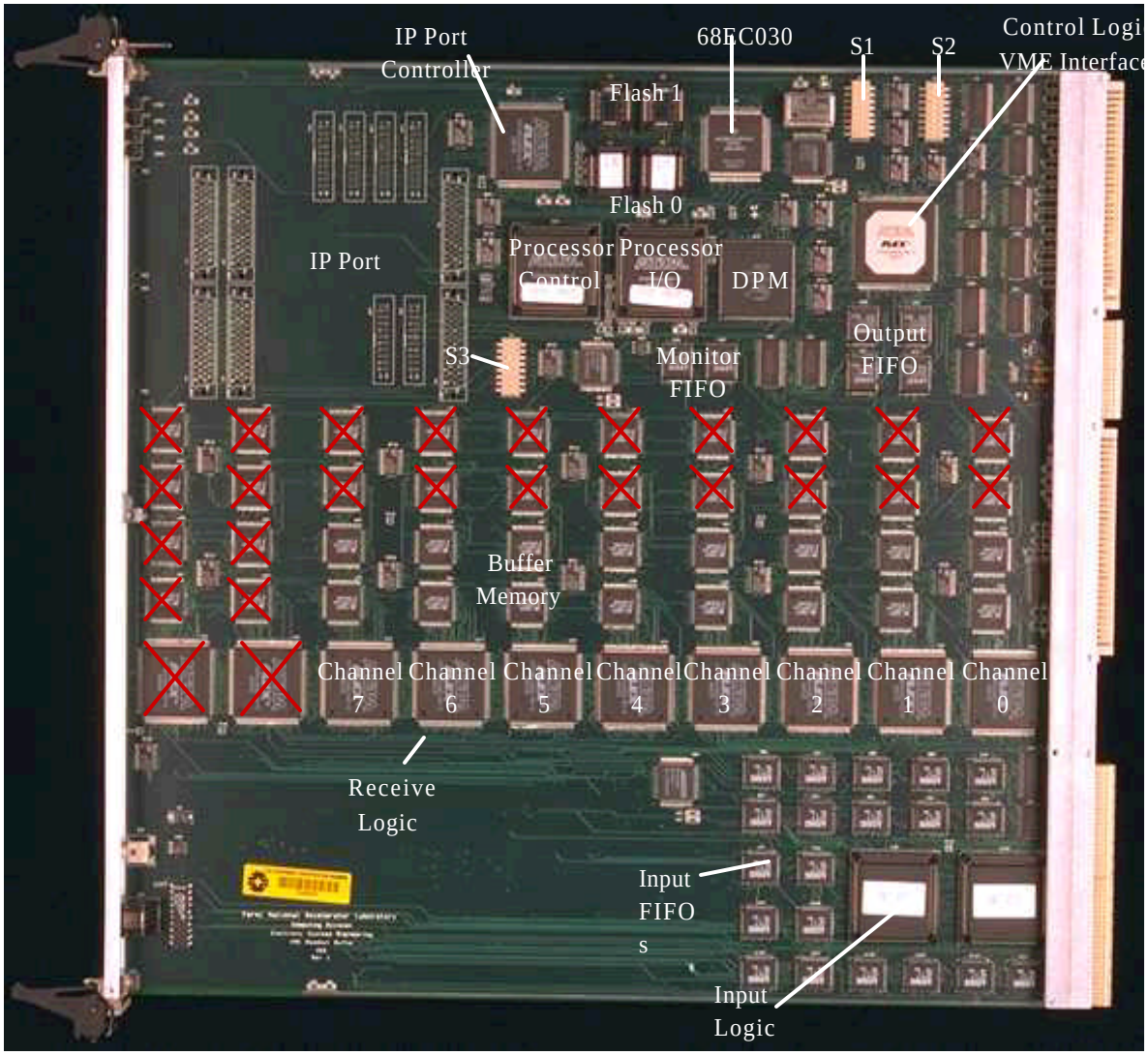
Normal S3 settings

for D0 SVX mode



flash bank 0 write protect
flash bank 1 write protect
diagnostic port disable
remote mode change disable
application select 0
application select 1
application select 2
application select 3

The chip locations in red are not populated on the D0 version of the VRB. Readout channel ordering is as shown.



D0 SVX VRB

Control of the VRB modules is provided by the VRBC. Following a Level 1 Accept, the VRBC Supplies the VRB with a buffer number for data input. Following a Level 2 Accept, the VRBC supplies the VRB with a buffer number for data output to VME.

SVX event data is logically organized by words (2 bytes). The first bytes of each data stream contain a header inserted by the front-end electronics to identify the data source (?). This is followed by a block of data from each SVX IC containing the chip ID, status and up to 128 channel #/data pairs. The record is terminated by an even number of end-of-record characters;

Sequencer ID
Sequencer Status
Chip ID
00
Channel #
Data
Channel #
Data
:
Chip ID
Status
Channel #
Data
:
EOR
EOR

Because the data streams in each link are independent, the transferred number of bytes can be different and two streams sharing the same link are required to provide separate End-of-record (EOR) signals. The VRB input FIFO will accept data from a G-link while the G-Link Data Valid signal is asserted. The Receive Logic will process the data until it recognizes an EOR character (any character with a high byte containing a value of "C0"). It will then discard data until the input FIFO is empty. When all the channels transmitting data to a VRB are done, the VRB will inform the VRBC. This is accomplished by releasing the READOUT_BUSY* signal on the J5/6 backplane. READOUT_BUSY* is an open collector signal that can be driven low by any of the VRBs in the

subrack. The event readout at the subrack level is finished when all VRBs have released this line. When readout is complete, it takes a few microseconds to save the event byte count and status information before the VRB releases it's READOUT_BUSY* signal and the event is available for scan.

The beginning of data input is initiated by the VRBC when it sends a READOUT BUFFER NUMBER message to the VRB. Any data received before the READOUT BUFFER NUMBER message will be held in the input FIFOs (up to the 512 byte limit).

Data received on each link is stored in the buffer pointed to by the Readout Buffer Number supplied by the VRBC.

Events that are accepted by the Level 2 trigger are copied to the VME output data FIFO (the VRBC supplies the SCAN BUFFER number). For events rejected by the Level 2 trigger, the VRBC simply re-uses the buffer number, causing the previous event data to be overwritten.

In this application, the G-links are used in 16 bit mode and each transfer packs data from two SVX HDIs. A total of eight HDI channels are supported. Each channel of Receive Logic processes 8 bits of data.

VRBC Interface

Messages received from the VRBC include the following;

- Message type 1: READOUT BUFFER NUMBER.....received after a Level 1 Accept to designate the next buffer number for VRB input.
- Message type 2: (not used)
- Message type 3: BUNCH CROSSING NUMBER.....received after a Level 1 Accept and used to check consistency with values in the data stream.
- Message type 4: SCAN BUFFER NUMBER.....received after a Level 2 Accept to designate the next buffer number for VRB output.
- Message type 5: EVENT NUMBER.....received after a Level 2 Accept and inserted into the output event record to label the Level 2 event.
- Message type 6-12: (not used)....available for expansion.
- Message type 13: CLEAR ERRORS.....clear any error signals on the status bus and status registers without resetting the VRB.
- Message type 14: RESET/RESTART.....a "reset" command causes reset of the input FIFOs and resynchronization of the Receive Logic to the start of the next event. RESET is necessary when changing VRB channel enable or emulation mode registers. A "restart" command reininitializes the entire VRB, including download of FPGAs.

The 12 bit message is interpreted as a four bit field (11:8) identifying the message type and an eight bit field (7:0) containing the message value.

The VRB drives the following status signals on the J5/6 backplane. These signals are active low (open collector TTL). The status signals returned to the VRBC are the OR of all VRB modules in the subrack and should be synchronized at the Controller:

- READOUT_BUSY* (stat0): Indicates that all data from the current input event has been received. READOUT_BUSY* is driven low when a READOUT BUFFER NUMBER message is received by the VRB and is released when all active channels on the VRB have received the event data. Note: this protocol does not allow pipelining of READOUT BUFFER NUMBER messages....the VRBC must wait until READOUT_BUSY* is high before sending the next READOUT BUFFER NUMBER message.
- SCAN_BUSY* (stat1): Indicates that all data from the current output event has been read from the VRB. SCAN_BUSY* is driven low when a SCAN BUFFER NUMBER message is received by the VRB and is released when all data for that event has been read from the VRB Output FIFO via VME. Note: this protocol does not allow pipelining of SCAN BUFFER NUMBER messages....the VRBC must wait until SCAN_BUSY* is high before sending the next SCAN BUFFER NUMBER message.
- SYNC_ERROR* (stat2): Indicates that a G-Link synchronization error was detected on a data link connected to an active channel. This signal remains asserted only while the sync error exists.
- (stat7-3) (reserved for error status signals):
- SCAN_READY (stat8): Indicates that data from the current output event is being copied to the output FIFO and may be read by the VBD. SCAN_READY is driven low when a SCAN BUFFER NUMBER message is received by the VRB and returns high when the event header has been written to the output FIFO.
- (stat9) (reserved):

Output Data Format

The following header information is added to each D0 SVX event:

<table>
<tr><td colspan="3">31 24 23 16 15 8 7 0
total byte count</td></tr>
<tr><td>User Info
(see "User Info" VME register)</td><td>Module ID
(slot number)</td><td>event number</td></tr>
<tr><td>Firmware Version Number
(date code)</td><td colspan="2">Configuration Info
bit 0 = gray decoding enabled
bit 1 = Trigger mode (always 0 for SVX record)</td></tr>
</table>

Status Word (TBD, currently all zeros)	
channel 0 byte count	channel 1 byte count
channel 2 byte count	channel 3 byte count
channel 4 byte count	channel 5 byte count
channel 6 byte count	channel 7 byte count

The VBD reads the VRB scan byte count or scan word count register and then performs a block transfer from the VRB output FIFO. The scan byte count register contains the byte count of the last event for which a SCAN BUFFER NUMBER message was processed.

Scan Processing

A typical scan operation is a two step process;

1) read the "scan byte count" register (this value should be the same as the "total byte count" which appears in the first header word of the event)

2) use the "scan byte count" to read the event from the Output data FIFO (using VME block transfer mode)

The event header is 32 bytes long and the data from each channel is padded to an eight byte boundary. The "scan byte count" includes all header, data and padding.

The byte count for each individual channel represents only the actual data for that channel (it does not include any padding bytes at the end of the channel's data). To find the start of the next channel's data, round the byte count for the previous channel to the next eight byte boundary.

Because the scan byte count register is simply a copy of the total byte count appearing in the first header word of the event, the first word of the event header can also be used to determine the size of the block transfer. The block transfer would then be 4 bytes smaller since the first header word has already been read. The advantage of this approach is that event scans can be pipelined to eliminate the scan command processing overhead of the first VRB. If the scan byte count register is used, the scans must be sequential to avoid overwriting the register.

Restart Operation

A VRB restart is generated under the following conditions;

1) power ON

2) front panel restart button

3) VME SYSRESET signal

4) expiration of the VRB processor watchdog timer (if enabled)

5) a VME or VRBC remote restart command (if enabled)

"Restart" causes the VRB processor to reset which in turn reloads all FPGA programs for the selected application. The processor reinitializes shared memory parameters and clears the input and output data FIFOs. The restart process takes several seconds to download FPGA code. Completion of restart is indicated by the front panel Processor LED.

The VRB will also respond to various reset commands which can be transmitted by the VRBC or through VME (VRB Reset/Restart register). These commands are only recognized if the processor and VME interface FPGA programs are functioning correctly.

VME Addressing

The D0 SVX application uses either A24 or A32 addressing with the following address modifiers;

0x39 or 0x3D (A24 single word accesses)

0x3B or 0x3F (A24:D32 block transfers)

0x38 or 0x3C (A24:D64 block transfers)

0x09 or 0x0D (A32 single word accesses)

0x0B or 0x0F (A32:D32 block transfers)

0X08 or 0x0C (A32:D64 block transfers)

The only VRB address that responds to block transfers is the output data FIFO.

To conserve FPGA resources, D8 transfers are not currently implemented (as required by the VME standard).

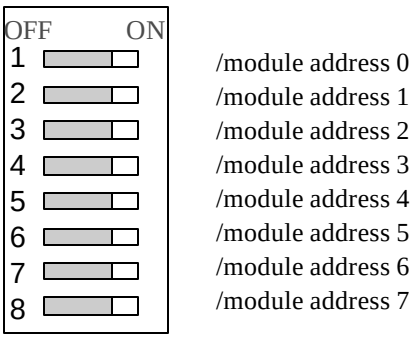
The base address is determined by a combination of subrack geographical address (GA4:0) and address select switches. This address decoding scheme is a result of backward compatibility requirements in the D0 system.

Address	**A24 Decoding**	**A32 Decoding**
A31	x	x
A30	x	x
A29	x	x
A28	x	x
A27	x	x
A26	x	x

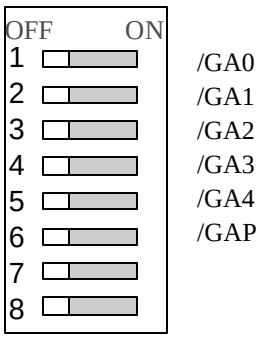
A25	x	x
A24	x	x
A23	address switch S1-8	address switch S1-8
A22	address switch S1-7	address switch S1-7
A21	address switch S1-6	address switch S1-6
A20	GA4	GA4
A19	GA3	GA3
A18	GA2	GA2
A17	GA1	GA1
A16	GA0	GA0

Switch bank S1 determines the high order bits of the module base address (ON = 0, OFF = 1). The D0 SVX application uses only three of these switches (S1-6,7,8). The position of switches S1-1,2,3,4,5 are "don't care". Switch bank S2 provides a means of setting the geographical address if geographical addressing is not supported by the subrack backplane (ON = 0, OFF = 1). For a standard VIPA backplane, these switches should all be OFF.

S1



S2



All registers, except the Output data FIFO, are accessed using D16 single word transfers. The Output data FIFO is accessed using D32 or D64 block transfers. Addresses not listed are reserved. Registers that are written by the VRB and should be considered "read-only" for VME access are indicated by *.

VME Address (hex)	**Function**
00000000	*VRB ID
00000002	*Configuration Setting
00000004	*Date code

00000006	*Module serial number
00000008	*Module type
0000000C	*Reset Code
0000000E	User Info
00000010	*Output data FIFO (non-pipelined mode) [d16 (upper half of 32 bit word), d32, BLT]
00000012	*Output data FIFO (non-pipelined mode) [d16 (lower half of 32 bit word)
00000018	*Output data FIFO (pipelined mode) [BLT, MBLT]
00000022	Readout buffer number
00000026	Readout bunch crossing number
00000028	Scan buffer number
0000002A	Scan event number
00000030	*Scan byte count
00000032	*Scan word count
00000034	*Scan longword count
00000038	*Current status
0000003A	Latched status (clear status)
0000003C	VRB reset/restart
00000040	VRB control
00000042	VTM control
00000044	*Monitor status
00000046	*Monitor count
00000048	Monitor channel
0000004A	Monitor control
0000004C	*SVX header 1 (histogram mode)
0000004E	*SVX header 2 (histogram mode)
00000050	*Chip 0 ID/status (histogram mode)
00000052	*Chip 1 ID/status (histogram mode)
00000054	*Chip 2 ID/status (histogram mode)
00000056	*Chip 3 ID/status (histogram mode)
00000058	*Chip 4 ID/status (histogram mode)
0000005A	*Chip 5 ID/status (histogram mode)
0000005C	*Chip 6 ID/status (histogram mode)
0000005E	*Chip 7 ID/status (histogram mode)
00000060	*Chip 8 ID/status (histogram mode)
00000062	*Chip 9 ID/status (histogram mode)
00000064	*Chip 10 ID/status (histogram mode)
00000066	*Chip 11 ID/status (histogram mode)
00000068	*Chip 12 ID/status (histogram mode)
0000006A	*Chip 13 ID/status (histogram mode)

0000006C	*Chip 14 ID/status (histogram mode)
0000006E	*Chip 15 ID/status (histogram mode)
00000070	Channel enable
00000072	Emulation mode enable
00000076	Gray code enable
00000078	*VTM link 0 power
0000007A	*VTM link 1 power
0000007C	*VTM link 2 power
0000007E	*VTM link 3 power
00000080-0000008E	*Channel control words (channel 0-7)
000000A0-000000AE	Channel status enables (channel 0-7)
00000300-0000037E	Buffer Start Address (buffer 0-63)
00000380-000003FE	Buffer Size (buffer 0-63)
00000800-00001FFE	*Monitor data/histogram buffer
00002000-0000207E	*Channel 0 byte counts (buffer 0-63)
00002080-000020FE	*Channel 1 byte counts (buffer 0-63)
00002100-0000217E	*Channel 2 byte counts (buffer 0-63)
00002180-000021FE	*Channel 3 byte counts (buffer 0-63)
00002200-0000227E	*Channel 4 byte counts (buffer 0-63)
00002280-000022FE	*Channel 5 byte counts (buffer 0-63)
00002300-0000237E	*Channel 6 byte counts (buffer 0-63)
00002380-000023FE	*Channel 7 byte counts (buffer 0-63)
00002500-0000257E	*Total byte count (buffer 0-63)
00002800-0000287E	*Channel 0 status (buffer 0-63)
00002880-000028FE	*Channel 1 status (buffer 0-63)
00002900-0000297E	*Channel 2 status (buffer 0-63)
00002980-000029FE	*Channel 3 status (buffer 0-63)
00002A00-00002A7E	*Channel 4 status (buffer 0-63)
00002A80-00002AFE	*Channel 5 status (buffer 0-63)
00002B00-00002B7E	*Channel 6 status (buffer 0-63)
00002B80-00002BFE	*Channel 7 status (buffer 0-63)

Register Definitions

VRB ID

Always 3 for a VRB module.

Configuration Setting

This register displays the current selected VRB application. For the D0 SVX application, the value should be 0.

Date Code

The version of the VRB firmware for the application currently running is indicated by a 4 digit hexadecimal date code. The format is YMDD where;

Y = last digit of year

M = month (January = “1”.....December = “C”)

DD = day

Module Serial Number

Individual board serial number for this VRB.

Module Type

Contains a value of 1 if this is a 10 channel X 64 Kbyte board, 0 if this is an 8 channel X 32 Kbyte board (normal D0 configuration).

User Info

This is a 16 bit user definable register which is inserted into the header of every event.

Readout Buffer Number

The VRB buffer number (6 bits) to use for readout of the current event. This information is normally supplied by the VRBC through the system controller interface, but may also be written to this register through VME for test purposes. The VRB will return READOUT_BUSY when the Readout Buffer Number is received, but will not begin processing the readout until both the Readout Buffer Number and Readout Bunch Crossing Number are received.

Readout Bunch Crossing Number

The bunch crossing number (8 bits) to use for consistency checking during readout of the current event. This information is normally supplied by the VRBC through the system controller interface, but may also be written to this register through VME for test purposes.

Scan Buffer Number

The VRB buffer number (6 bits) to use for output of the next event to the scan processor. This information is normally supplied by the VRBC through the system controller interface, but may also be written to this register through VME for test purposes. The VRB will return SCAN_BUSY when the Scan Buffer Number is received, but will not begin processing the scan until both the Scan Buffer Number and Scan Event Number are received.

Scan Event Number

The event number (8 bits) to be inserted into the header field of the next scanned event. This information is normally supplied by the VRBC through the system controller interface, but may also be written to this register through VME for test purposes.

Scan Byte Count

This register contains the scan byte count for the VRB buffer designated in the most recent Scan Buffer Number message from the VRBC. The scan byte count is the total of VRB data from all active channels plus header. This register is updated when the scan operation is complete (SCAN_BUSY signal released). Prior to completion of the scan operation, this register will contain the byte count for the previously scanned event.

Scan Word Count

This register contains the scan word count for the VRB buffer designated in the most recent Scan Buffer Number message from the VRBC. This is equivalent to the value in the Scan Byte Count register divided by two. The scan word count is the total of VRB data from all active channels plus header.

Scan Longword Count

This register contains the scan longword count for the VRB buffer designated in the most recent Scan Buffer Number message from the VRBC. This is equivalent to the value in the Scan Byte Count register divided by four. The scan longword count is the total of VRB data from all active channels plus header.

Current Status

This register holds an unlatched copy of the 10 status signals driven on the VRB subrack backplane to the VRBC. Error signals are normally asserted at the completion of readout and remain asserted until the next readout command is received.

The SRC receives the OR of the status signals from all VRBs in the system. The VRB generating a particular error signal can be identified by examining this register individually in each VRB.

Latched Status

This register holds a latched copy of the 10 status signals driven on the VRB subrack backplane to the VRBC. If any of the status signals have been asserted since the last clear status command, the corresponding bit will be latched high in this register. Note that the READOUT_BUSY and SCAN_BUSY signals will be asserted in normal operation....these bits will always be latched high if the VRB has received at least one READOUT or SCAN command.

VRB Reset/Restart

The VRB can be programmed to recognize up to 256 reset/restart commands. These commands can be sent either through VME or by the VRBC. The following commands are currently implemented;

0x00	reset input and output data FIFOs, VRB control and buffer logic (does not reset processor or reload FPGAs)
0x01	same as 0, but also resets VTM
0x02-0x7F	(reset) not defined
0x80	restart VRB with application 0
0x81	restart VRB with application 1
0x82	restart VRB with application 2
0x83	restart VRB with application 3
0x84	restart VRB with application 4
0x85	restart VRB with application 5
0x86	restart VRB with application 6

0x87	restart VRB with application 7
0x88	restart VRB in Default Mode
0x89-0xFF	(restart) not defined

The restart commands are provided to allow test software to switch between applications without removing the VRB and changing default switch settings. This overrides the settings of the VRB Application Select switches (S3-5,6,7,8) until the next restart. Note that restart commands are only recognized if the Remote Restart Disable switch (S3-4) is OFF.

Channel Enable

The VRB data channels may be individually enabled or disabled. Disabled channels have a byte count of zero in the event header and no data in the data record. Error signals (loss of sync, etc) are not processed if the channel is not enabled. The default startup setting for this register is all channels disabled (0). For all eight channels enabled, the correct value for this register would be 0x00FF.

Important: settings for the channel enable register do not take effect until the VRB is reset by writing to the VRB reset/restart register (address 3C, reset command 0 or 1) or by sending a VRBC reset command (command 14, value 0 or 1).

Bit number	Function
0	channel 0 enable
1	channel 1 enable
2	channel 2 enable
3	channel 3 enable
4	channel 4 enable
5	channel 5 enable
6	channel 6 enable
7	channel 7 enable
8-15	not used

Emulation Mode Enable

The VRB data channels may be individually placed in “real” or “emulated” data modes. In emulated data mode, a 256 byte incrementing pattern is generated in place of input data from the SVX Sequencer for the selected channel(s). The generated data pattern increments from 0x00 to 0x7F with

the same count appearing in both halves of each 16 bit “SVX” word. The default startup setting for this register is all channels in “real” data mode (0).

Important: settings for the emulation mode enable register do not take effect until the VRB is reset by writing to the VRB reset/restart register (address 3C, reset command 0 or 1) or by sending a VRBC reset command (command 14, value 0 or 1).

Bit number	Function
0	channel 0 emulation mode
1	channel 1 emulation mode
2	channel 2 emulation mode
3	channel 3 emulation mode
4	channel 4 emulation mode
5	channel 5 emulation mode
6	channel 6 emulation mode
7	channel 7 emulation mode
8-15	not used

Gray Code Enable

This register contains a single bit which enables (1) or disables (0) gray decoding of the incoming SVX data bytes. Gray decoding does not take place for Sequencer id/status, SVX chip ID, SVX address or "virtual SVX" data.

Important: settings for the Gray code enable register do not take effect until the VRB is reset by writing to the VRB reset/restart register (address 3C, reset command 0 or 1) or by sending a VRBC reset command (command 14, value 0 or 1).

Bit number	Function
0	decoder enable
2-15	not used

VRB Control

This register is used to control some global features of the VRB.

Bit number	Function
0	Control Port enable
1	Status Signal enable
2-15	not assigned

The "control port enable" bit allows the user to turn off the VRB control port (J5/6). When OFF, the VRB will not respond to VRBC control messages. This bit is ON by default.

The "status signal enable" bit allows the user to turn off the VRB status signals (J5/6). When OFF, the VRB drives all (active low) backplane status and error signals to a high state. This bit is ON by default.

Both of these control bits should be cleared when the VRB is not being used in the system. This will disable VRB - VRBC interactions for this board. The VRB will continue to respond to VME commands and will update the VME status register.

VTM Control

A command byte written to this register will be sent directly to the VTM for processing. The command format is as follows;

Bit number	Function
0	Read optical power level (A/D)
1	Read optical signal detect
2	Disable GLink clock
3	Enable GLink clock
4	Reset GLink
5-6	VTM Channel number (0-3)
7-15	not used

This register will be reset to zero as soon as the command is executed. Only ONE of the command bits (bits 0-4) may be set. If more than one bit is set, only the highest priority command will be executed (bit 0 = highest.....bit 4 = lowest priority).

Examples;

1) Read the optical power level on link 2

VTM Control Register = 0x0041

2) Reset link 3

VTM Control Register = 0x0070

Monitor Count

Returns the number of bytes in the monitor buffer for a monitor snapshot operation or the number of events histogrammed for a monitor histogram operation. In snapshot mode, the count is updated after the data has been copied from the monitor FIFO to the monitor buffer (VME memory). In histogram mode, the count is updated continuously for each event histogrammed.

Monitor Channel

Used to set the VRB channel number (0-7) being monitored.

Monitor Control

This register determines the type of monitor operation performed. Defined operations are;

0	Idle	(no operation)
1	Reset	Clear the monitor buffer and reset the Monitor Count register. (the value in the Monitor Control register returns to 0 when the reset operation is complete)
2	Snapshot	Copy data from a single event to the monitor buffer. (the value in the Monitor Control register returns to 0 when the snapshot operation is complete)
3	Histogram	Generate a histogram of SVX channels (the histogram operation continues until the Monitor Control register is set back to 0 by the user.) The histogram table appears in the monitor buffer with the following order; Locations 0x800- 0x8FE SVX chip 0, channels 0-127 Locations 0x900- 0x9FE SVX chip 1, channels 0-127 ….. Locations 0x1700-0x17FE SVX chip 15, channels 0-127 (Note that SVX chips are ordered by their position in the data stream, not by the SVX chip ID. Channel 127 is not histogrammed if the SVX chip ID is 0.)

VTM Power

After transferring control to the D0 SVX application the VRB issues a sequence of commands to the VTM (via the VTM serial port) to read the Finisar optical receiver power. The measured power levels for links 0,1,2 and 3 are stored in separate registers (00000078 through 0000007E, respectively). The low 8 bits of the register contain the power value read by the A/D converter. Bit 8 of the register is set if the Finisar optical receiver "signal detect" is set. If this is not set, the fiber optic cable may be disconnected or damaged. The power measurement and signal detect status are only checked once at startup or in response to a VTM control register write.....they are not continuous readings.

Buffer Start Address (0-63)

Each VRB data channel has a total of 32 KBytes of buffer memory. This buffer memory is divided into one or more fixed size event buffers. The VRB control logic allows up to 64 buffers. The starting address of each of the VRB event buffers may be individually programmed. Starting addresses for each buffer are the same for all eight VRB data channels. The default setting for the D0 SVX configuration is two overlapping sets of buffers; set **1** is 16 buffers of 2 KBytes each (buffer numbers 0-15) and set **2** is 8 buffers of 4 KBytes each (buffer numbers 16-23).

Buffer Number	**Register Address (hex)**
0	00000300
1	00000302
2	00000304
3	00000306
...	...
63	0000037E

The following examples are for a system with 8 buffers;

Buffer Number	**Starting Address (hex)**	**Buffer Size**	**Comments**
0	0000	4 KBytes	8 equal sized buffers
1	1000	4 KBytes	
2	2000	4 KBytes	

3	3000	4 KBytes	
4	4000	4 KBytes	
5	5000	4 KBytes	
6	6000	4 KBytes	
7	7000	4 KBytes	

Buffer Number	**Starting Address (hex)**	**Buffer Size**	**Comments**
0	0000	2 KBytes	6 smaller buffers
1	0800	2 KBytes	
2	1000	2 KBytes	
3	1800	2 KBytes	
4	2000	2 KBytes	
5	2800	2 KBytes	
6	3000	10 KBytes	
7	5800	10 KBytes	and 2 larger buffers

The VRB logic places no restrictions on the starting address assigned to each buffer (buffers may overlap). The following example would allow operation in three different configurations without resetting the VRB. The configuration would be selected by using a different set of buffer numbers;

Configuration 1 buffer numbers 0-7

Configuration 2 buffer numbers 8-11

Configuration 3 buffer numbers 12-13

Buffer Number	**Starting Address (hex)**	**Buffer Size**	**Comments**
0	0000	4 KBytes	8 buffers of 4 KBytes each
1	1000	4 KBytes	
2	2000	4 KBytes	

3	3000	4 KBytes	
4	4000	4 KBytes	
5	5000	4 KBytes	
6	6000	4 KBytes	
7	7000	4 KBytes	
8	0000	8 KBytes	4 buffers of 8 KBytes each
9	2000	8 KBytes	
10	4000	8 KBytes	
11	6000	8 KBytes	
12	0000	16 KBytes	2 buffers of 16 KBytes each
13	4000	16 KBytes	

The default startup buffer assignment is shown in the following table;

Buffer Number	**Starting Address (hex)**	**Buffer Size**	**Comments**
0	0000	2 KBytes	16 buffers of 2 KBytes each
1	0800	2 KBytes	
2	1000	2 KBytes	
3	1800	2 KBytes	
4	2000	2 KBytes	
5	2800	2 KBytes	
6	3000	2 KBytes	
7	3800	2 KBytes	
8	4000	2 KBytes	
9	4800	2 KBytes	
10	5000	2 KBytes	
11	5800	2 KBytes	

12	6000	2 KBytes	
13	6800	2 KBytes	
14	7000	2 KBytes	
15	7800	2 KBytes	
16	0000	4 KBytes	8 buffers of 4 KBytes each
17	1000	4 KBytes	
18	2000	4 KBytes	
19	3000	4 KBytes	
20	4000	4 KBytes	
21	5000	4 KBytes	
22	6000	4 KBytes	
23	7000	4 KBytes	

Buffer Size (0-63)

Each VRB data channel has a total of 32 KBytes of buffer memory. This buffer memory is divided into one or more fixed size event buffers. The VRB control logic allows up to 64 buffers. The size of each of the VRB event buffers may be individually programmed and is the same for all ten VRB data channels. The current startup default setting for the D0 SVX configuration is 16 buffers of 2 KBytes each.

The buffer size parameter is used only to determine buffer overflow error. The VRB will truncate event data written to a buffer when the byte count exceeds the Buffer Size parameter for that buffer. Because buffer start addresses are not necessarily consecutive and may overlap, the buffer size is not automatically calculated from the buffer start addresses.

Buffer Number	**Register Address (hex)**
0	00000380
1	00000382
2	00000384
3	00000386

...	...
63	000003FE

Output Data Interface

The output data FIFO can be read in two ways (using separate VME addresses). The "pipelined" mode is much faster, but has a minor access restriction.

To maximize transfer rate, the "pipelined" mode prefetches data from the output data FIFO and places it in pipeline registers. When the VRB receives a VME data strobe (DS), it can immediately transfer the contents of the last pipeline register to the VME data bus and return DTACK within 20 ns. Once the VRB begins a pipelined block transfer, it assumes that the Scan Processor will complete the block transfer before accessing other registers in the VRB. The block transfer may be interrupted by other VME activity, as long as it is not a random read/write operation to the specific VRB involved in the block transfer. If this happens, the last two data words which were prefetched from the output data FIFO before the block transfer was interrupted will be discarded.

The "non-pipelined" block transfer mode does not prefetch data and so does not have the same restriction, but is 2-3 times slower. D64 block transfers (MBLT) only operate in the pipelined mode.

Application 1 (CDF SVX)

In the CDF SVX application, the VRB acts as a buffer for data pending a Level 2 trigger decision. Buffer management is provided by a single controller (SRC) which resides in one of the VRB subracks and communicates with all VRB modules through the VRB Fanout module located in slot 14 of each subrack. A subrack may contain up to 12 VRB modules. Each VRB receives data from five detector layers via a "Fiber Interface Board" and G-Link VTM transition module.

The VRB startup application and some programming/diagnostic features are controlled by switch bank S3;

S3-1 and S3-2 are write protect switches for the VRB flash memory. These may normally be left OFF (not write protected) to enable in-system updates of the VRB firmware and software. Since the VRB software does not allow flash write cycles when in CDF SVX mode, there is generally no danger of accidentally overwriting the flash memory.

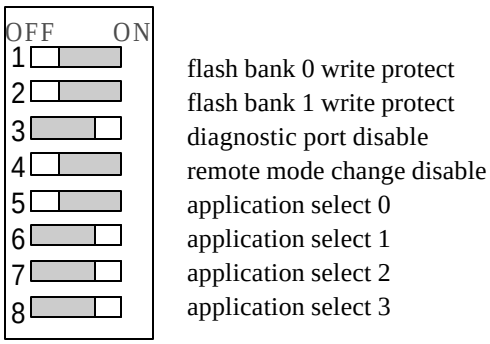
Switch S3-3 disables the RS-232 terminal diagnostic port. This port outputs information at VRB startup/reset and during some error conditions. If there is no terminal connected, this switch should be ON (disabled), since the VRB takes longer to recognize and execute a reset command when it is displaying diagnostic messages.

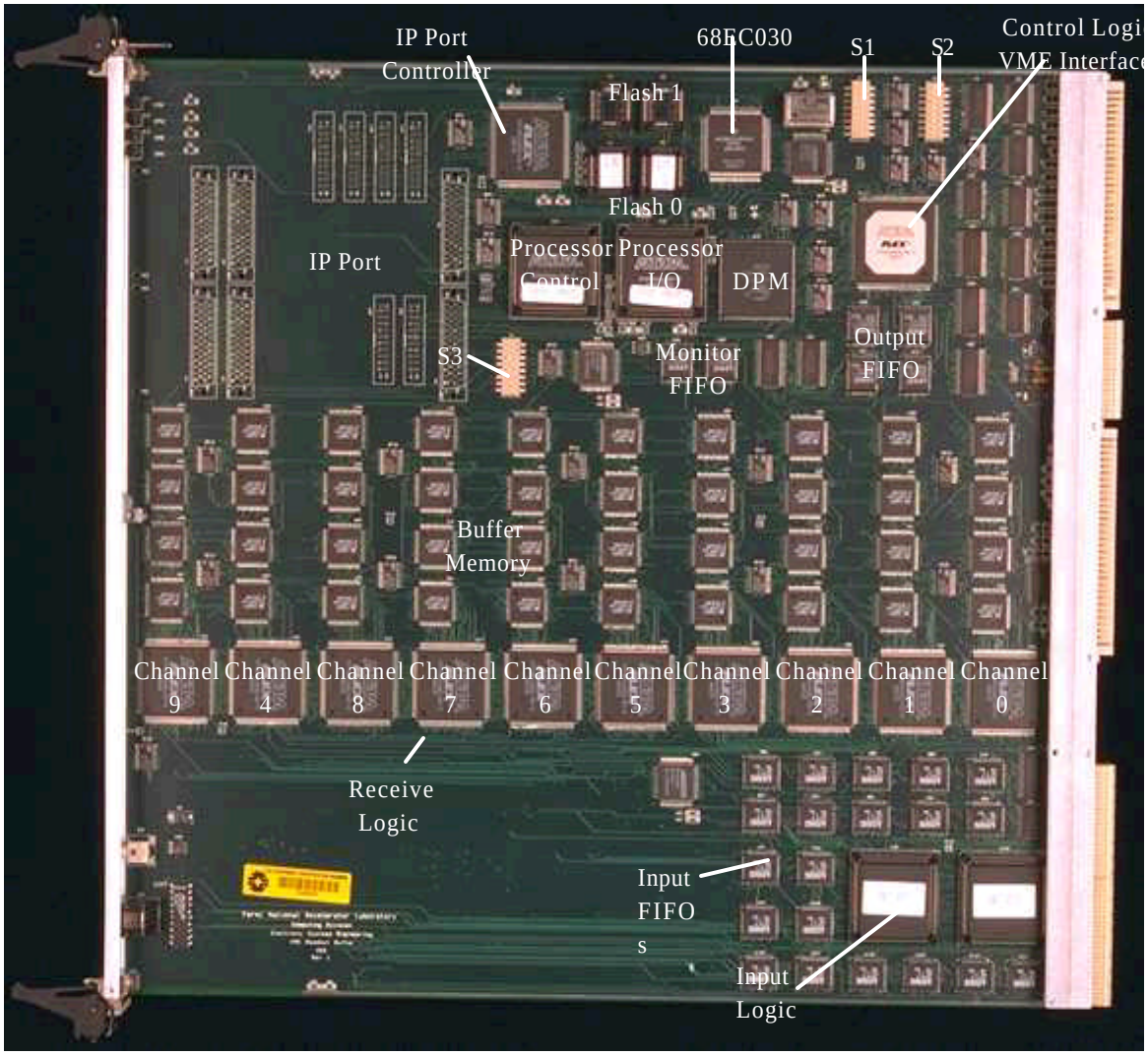
Switch S3-4 disables the remote mode change feature. When enabled (switch OFF), the VRB can be remotely rebooted into a mode other than CDF SVX. This is necessary for remote reprogramming.

Switches S3-5 through S3-8 are used to select the VRB startup application. Application select switch settings for CDF SVX mode are as follows;

	S3-8	**S3-7**	**S3-6**	**S3-5**
CDF SVX	ON	ON	ON	OFF

Normal S3 settings for CDF SVX mode





CDF SVX VRB

Control of the VRB modules is provided by the Silicon Readout Controller (SRC). Following a Level 1 Accept, the SRC supplies the VRB with a buffer number for data input. Following a Level 2 Accept, the SRC supplies the VRB with a buffer number for data output to VME.

SVX event data is logically organized by words (2 bytes). The first bytes of each data stream contain a header inserted by the front-end electronics to identify the data source. This is followed by a

block of data from each SVX IC containing the chip ID, status and up to 128 channel #/data pairs. The record is terminated by an even number of end-of-record characters;

IHDR0
IHDR1
IHDR2
IHDR3
Chip ID
Status
Channel #
Data
Channel #
Data
:
Chip ID
Status
Channel #
Data
:
EOR
EOR
EOR
EOR

Because the data streams in each link are independent, the transferred number of bytes can be different and three streams sharing the same link are required to provide separate End-of-record (EOR) signals. The VRB input FIFO will accept data from a G-link when the G-Link Data Valid signal is asserted. The Receive Logic will process the data until it recognizes an EOR character (any character with a high byte containing a value of "CX"). It will then discard data until the input FIFO is empty. Prior to receiving the EOR character, the VRB will also discard any 2 byte SVX data word that contains the value "D0" in the high byte. This is used as a fill character by the FIB to avoid data underrun.

When all the channels transmitting data to a VRB are done, the VRB will inform the Controller. This is accomplished by releasing the READOUT_BUSY* signal on the J5/6 backplane. READOUT_BUSY* is an open collector signal that can be driven low by any of the VRBs in the subrack. The event readout at the subrack level is finished when all VRBs have released this line. When readout is complete, it takes a few microseconds to save the event byte count and status information before the VRB releases it's READOUT_BUSY* signal and the event is available for scan.

The VRB Receive Logic checks the event data as it is being received. The following is a list of the checks performed for SVX data:

- check that the pipeline capacitor number and the bunch crossing number match values received from the SRC
- on receipt of EOR, append EOR bytes to the event until the event size per channel is a multiple of 8 bytes. This is done to simplify VME block transfers.
- if the readout is truncated due to buffer overflow or format error, a special EOR code is appended.

The beginning of data input is initiated by the SRC when it sends a READOUT BUFFER NUMBER message to the VRB. Any data received before the READOUT BUFFER NUMBER message will be held in the input FIFOs (up to the 512 byte limit).

Data received on each link is stored in the buffer pointed to by the Readout Buffer Number supplied by the SRC.

Events that are accepted by the Level 2 trigger are copied to the VME output data FIFO (the SRC supplies the SCAN BUFFER number). For events rejected by the Level 2 trigger, the SRC simply re-uses the buffer number, causing the previous event data to be overwritten.

In this application, the G-links are used in 20 bit mode and each transfer packs data from two and a half SVX HDIs. A total of ten HDI channels are supported. Each channel of Receive Logic processes 8 bits of data.

SRC Interface

Control messages received from the SRC include the following;

- Message type 1: READOUT BUFFER NUMBER.....received after a Level 1 Accept to designate the next buffer number for VRB input.
- Message type 2: PIPELINE CAPACITOR NUMBER.....received after a Level 1 Accept and used to check consistency with values in the data stream.
- Message type 3: BUNCH CROSSING NUMBER.....received after a Level 1 Accept and used to check consistency with values in the data stream.
- Message type 4: SCAN BUFFER NUMBER.....received after a Level 2 Accept to designate the next buffer number for VRB output.
- Message type 5: EVENT NUMBER.....received after a Level 2 Accept and inserted into the output event record to label the Level 2 event.
- Message type 6: TSL1 .....received after a Level 1 Accept and inserted into the output event record to indicate “time since Level 1 Trigger”.
- Message type 7-12: (not used)....available for expansion.

- Message type 13: CLEAR STATUS.....clear the VRB status registers and signals.
- Message type 14: RESET/RESTART.....a ”reset” command causes reset of the input FIFOs and resynchronization of the Receive Logic to the start of the next event. RESET is necessary when changing VRB channel enable or emulation mode registers. A “restart” command reininitializes the entire VRB, including download of FPGAs.

The 12 bit message is interpreted as a four bit field (11:8) identifying the message type and an eight bit field (7:0) containing the message value. The CONTROLLER_ERROR signal is asserted if a READOUT BUFFER NUMBER message is received while the VRB is still asserting READOUT_BUSY or if a SCAN BUFFER NUMBER message is received while the VRB is still asserting SCAN_BUSY.

The VRB drives the following status signals on the J5/6 backplane. These signals are active low (open collector TTL) on the backplane and are inverted by the VRB Fanout module. The status signals returned by the Fanout Module(s) to the SRC are the OR of all VRB modules in the system and should be synchronized at the SRC:

- READOUT_BUSY* (stat0): Indicates that all data from the current input event has been received. READOUT_BUSY* is driven low when a READOUT BUFFER NUMBER message is received by the VRB and is released when all active channels on the VRB have received the event data. Note: this protocol does not allow pipelining of READOUT BUFFER NUMBER messages....the SRC must wait until READOUT_BUSY* is high before sending the next READOUT BUFFER NUMBER message.
- SCAN_BUSY* (stat1): Indicates that all data from the current output event has been transmitted. SCAN_BUSY* is driven low when a SCAN BUFFER NUMBER message is received by the VRB and is released when all data for that event has been transferred to the VRB Output FIFO. Note: this protocol does not allow pipelining of SCAN BUFFER NUMBER messages....the SRC must wait until SCAN_BUSY* is high before sending the next SCAN BUFFER NUMBER message.
- SYNC_ERROR* (stat2): Indicates that a G-Link synchronization error or a non-valid G-Link data word (frame error) was detected on a data link connected to an active channel. This synchronization error remains asserted only while the sync error exists. The frame error signal is asserted following readout of the event (prior to READOUT_BUSY* high) and remains asserted until the next READOUT BUFFER NUMBER message is received.
- TIMEOUT* (stat3): Indicates that no data has been received on a VRB channel for 20 usec following a readout command.
- IDENTIFIER_ERROR* (stat4): Indicates that an invalid event identifier (crossing number or capacitor number) was detected in a data stream. This signal is asserted following

readout of the event (prior to READOUT_BUSY* high) and remains asserted until the next READOUT BUFFER NUMBER message is received.

- FORMAT_ERROR* (stat5): Indicates that a data format error has occurred. Data format errors indicate markers (channel numbers, chip IDs, etc.) which are outside the valid range, or not in the expected location in the data stream. This signal is asserted following readout of the event (prior to READOUT_BUSY* high) and remains asserted until the next READOUT BUFFER NUMBER message is received.
- CONTROLLER_ERROR* (stat6): Indicates that an invalid or unrecognizable SRC message was received. It is also set if the SRC sends a READOUT command while the VRB is asserting READOUT_BUSY or a SCAN command while the VRB is asserting SCAN_BUSY. This signal is asserted at the time that the message is received and remains asserted until a clear status command is received.
- VRB_ERROR* (stat7): Indicates that the VRB was not able to process the event correctly (usually due to buffer overflow). This signal is asserted following readout of the event (prior to READOUT_BUSY* high) and remains asserted until the next READOUT BUFFER NUMBER message is received.
- (stat8) (reserved):
- (stat9) (reserved):

Output Data Format

The following header information is added to each CDF SVX event:

<table>
<tr><td colspan="6">31 24 23 16 15 8 7 0</td></tr>
<tr><td colspan="3">total byte count</td><td colspan="2">tsl1</td><td>event number</td></tr>
<tr><td>ch0 stat (31..26)</td><td>ch1 stat (25..20)</td><td>ch2 stat (19..14)</td><td>ch3 stat (13..8)</td><td>ch4 stat (7..2)</td><td>00</td></tr>
<tr><td>ch5 stat (31..26)</td><td>ch6 stat (25..20)</td><td>ch7 stat (19..14)</td><td>ch8 stat (13..8)</td><td>ch9 stat (7..2)</td><td>00</td></tr>
<tr><td colspan="3">channel 0 byte count</td><td colspan="3">channel 1 byte count</td></tr>
<tr><td colspan="3">channel 2 byte count</td><td colspan="3">channel 3 byte count</td></tr>
<tr><td colspan="3">channel 4 byte count</td><td colspan="3">channel 5 byte count</td></tr>
<tr><td colspan="3">channel 6 byte count</td><td colspan="3">channel 7 byte count</td></tr>
<tr><td colspan="3">channel 8 byte count</td><td colspan="3">channel 9 byte count</td></tr>
</table>

The channel status words are mapped as follows (see previous section on SRC interface for error definitions);

Bit	First Status Word	Second Status Word
0	not used (0)	not used (0)
1	not used (0)	not used (0)
2	G-Link sync error (channel 4)	G-Link sync error (channel 9)
3	Frame error (channel 4)	Frame error (channel 9)
4	ID error (channel 4)	ID error (channel 9)
5	Format error (channel 4)	Format error (channel 9)
6	not used (0)	not used (0)
7	VRB error (channel 4)	VRB error (channel 9)
8	G-Link sync error (channel 3)	G-Link sync error (channel 8)
9	Frame error (channel 3)	Frame error (channel 8)
10	ID error (channel 3)	ID error (channel 8)
11	Format error (channel 3)	Format error (channel 8)
12	not used (0)	not used (0)
13	VRB error (channel 3)	VRB error (channel 8)
14	G-Link sync error (channel 2)	G-Link sync error (channel 7)
15	Frame error (channel 2)	Frame error (channel 7)
16	ID error (channel 2)	ID error (channel 7)
17	Format error (channel 2)	Format error (channel 7)
18	not used (0)	not used (0)
19	VRB error (channel 2)	VRB error (channel 7)
20	G-Link sync error (channel 1)	G-Link sync error (channel 6)
21	Frame error (channel 1)	Frame error (channel 6)
22	ID error (channel 1)	ID error (channel 6)
23	Format error (channel 1)	Format error (channel 6)
24	not used (0)	not used (0)
25	VRB error (channel 1)	VRB error (channel 6)

26	G-Link sync error (channel 0)	G-Link sync error (channel 5)
27	Frame error (channel 0)	Frame error (channel 5)
28	ID error (channel 0)	ID error (channel 5)
29	Format error (channel 0)	Format error (channel 5)
30	not used (0)	not used (0)
31	VRB error (channel 0)	VRB error (channel 5)

A trailer consisting of two zero words is added to the end of each event. If the output data FIFO is empty, subsequent read operations will return this last zero value. The VME Scan Processor can poll the output data FIFO until it returns a non-zero value. This will be the first word of the next scanned event. By polling the output data FIFO, the Scan Processor can operate independently of the SRC. Scanned events can be queued in the output data FIFO and read at the maximum rate of the Scan Processor. The Scan Processor does not need to know the scan operation status, event number or scan buffer number.

Scan Processing

The VRB is programmed to allow scan operations using a single VRB address (output data FIFO) and transfer mode (block transfer). A typical scan operation is a two step process;

1) read one 32 bit word from the output data FIFO (using VME block transfer mode).....repeat this step until a non-zero value is read.....this value will be the first word of the event
2) use the "total byte count" to read the remainder of the event (also using VME block transfer mode)

Note that the value of the "total byte count" includes itself, therefore the correct block size to transfer in step 2 is "total byte count" minus four.

This process can also be accomplished using 64 bit transfers. In that case, the first non-zero value returned will contain the first two 32 bit words of the event and the correct block size to transfer in step 2 is "total byte count" minus eight.

The event header is 32 bytes long and the data from each channel is padded to an eight byte boundary. The "total byte count" includes all header, data and padding.

The byte count for each individual channel represents only the actual data for that channel (it does not include any padding bytes at the end of the channel's data). To find the start of the next channel's data, round the byte count for the previous channel to the next eight byte boundary.

Reset/Restart Operation

The VRB will also respond to various reset commands which can be transmitted by the SRC or through VME (VRB Reset/Restart register). These commands are only recognized if the processor and VME interface FPGA programs are functioning correctly.

A VRB restart is generated under the following conditions;

1) power ON

2) front panel restart button

3) VME SYSRESET signal

4) expiration of the VRB processor watchdog timer

5) a VME or SRC remote restart command (if enabled)

"Restart" causes the VRB processor to reset, which in turn reloads all FPGA programs for the selected application. The processor reinitializes shared memory parameters and clears the input and output data FIFOs. The restart process takes several seconds to download FPGA code. Completion of restart is indicated by the front panel Processor LED.

A VRB reset requires less time since FPGA code is not downloaded. The processor will typically respond to a reset command within 100 microseconds.

VME Addressing

The CDF SVX application uses A32 addressing with the following address modifiers;

0x09, 0x0A, 0x0D or 0x0E (D32 single word access)

0x0B or 0x0F (D32 block transfers)

0X08 or 0x0C (D64 block transfers)

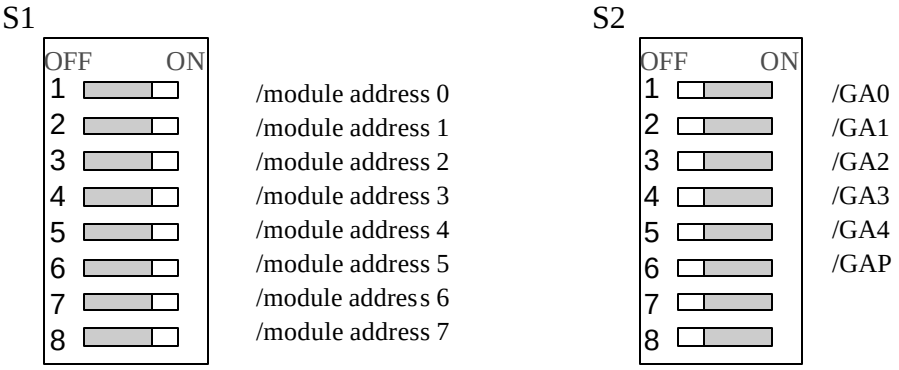
The only VRB location that responds to block transfers is the output data FIFO.

To conserve FPGA resources, D16 and D8 single word transfers are not currently implemented (as required by the VME standard).

The base address of the module is determined by the subrack geographical address (GA4:0).

Address	A32 Decoding
A31	GA4
A30	GA3
A29	GA2
A28	GA1
A27	GA0

There are two VME address configuration switches. These are normally not used in the CDF SVX mode, but must be set to the proper default positions. Switch bank S1 determines the eight bit module base address for systems not using geographical addressing (ON = 0, OFF = 1). For the CDF SVX application, these switches are "don't care" but are normally set to the ON position. Switch bank S2 provides a means of setting the base address if geographical addressing is used, but is not supported by the subrack backplane (ON = 0, OFF = 1). These switches should all be set to OFF in the CDF SVX application (when used with a standard VIPA subrack).



All registers, except the Output data FIFO, are accessed using D32 single word transfers. These registers contain 16 bits of significant information in the low half of the 32 bit word with the high 16 bits returning zeros on reads and ignored on writes. The Output data FIFO is accessed using D32 or D64 block transfers. Addresses not listed are reserved. Registers that are written by the VRB and should be considered "read-only" for VME access are indicated by *.

VME Address (hex)	Function
00000000	*VRB ID
00000004	*Application ID
00000008	*Date code
0000000C	*Module serial number

00000010	*Module type
00000014	*Configuration switches
00000018	*Reset code
00000044	Readout buffer number
00000048	Readout pipeline capacitor number
0000004C	Readout bunch crossing number
00000050	Scan buffer number
00000054	Scan event number
00000058	Time since L1
00000070	*Current status
00000074	Latched status (clear status)
00000078	VRB reset/restart
00000080	VRB control
00000084	VTM control
00000088	*Monitor status
0000008C	*Monitor count
00000090	Monitor channel
00000094	Monitor control
00000098	*SVX header 1 (histogram mode)
0000009C	*SVX header 2 (histogram mode)
000000A0	*Chip 0 ID/status (histogram mode)
000000A4	*Chip 1 ID/status (histogram mode)
000000A8	*Chip 2 ID/status (histogram mode)
000000AC	*Chip 3 ID/status (histogram mode)
000000B0	*Chip 4 ID/status (histogram mode)
000000B4	*Chip 5 ID/status (histogram mode)
000000B8	*Chip 6 ID/status (histogram mode)
000000BC	*Chip 7 ID/status (histogram mode)
000000C0	*Chip 8 ID/status (histogram mode)
000000C4	*Chip 9 ID/status (histogram mode)
000000C8	*Chip 10 ID/status (histogram mode)
000000CC	*Chip 11 ID/status (histogram mode)
000000D0	*Chip 12 ID/status (histogram mode)
000000D4	*Chip 13 ID/status (histogram mode)
000000D8	*Chip 14 ID/status (histogram mode)
000000DC	*Chip 15 ID/status (histogram mode)
000000E0	Channel enable
000000E4	Emulation mode enable
000000E8	WYSIWYG mode enable Emulation mode enable
000000EC	Timeout enable
000000F8	*VTM link 2 power

000000FC	*VTM link 3 power
00000100-00000124	*Channel control words (channel 0-9)
00000140-00000164	Channel status enables (channel 0-9)
00000600-000006FC	Buffer start address (buffer 0-63)
00000700-000007FC	Buffer size (buffer 0-63)
00001000-00003FFC	*Monitor data/histogram buffer
00004000-000040FC	*Channel 0 byte counts (buffer 0-63)
00004100-000041FC	*Channel 1 byte counts (buffer 0-63)
00004200-000042FC	*Channel 2 byte counts (buffer 0-63)
00004300-000043FC	*Channel 3 byte counts (buffer 0-63)
00004400-000044FC	*Channel 4 byte counts (buffer 0-63)
00004500-000045FC	*Channel 5 byte counts (buffer 0-63)
00004600-000046FC	*Channel 6 byte counts (buffer 0-63)
00004700-000047FC	*Channel 7 byte counts (buffer 0-63)
00004800-000048FC	*Channel 8 byte counts (buffer 0-63)
00004900-000049FC	*Channel 9 byte counts (buffer 0-63)
00004F00-00004FFC	*Total byte count (buffer 0-63)
00005000-000050FC	*Channel 0 status (buffer 0-63)
00005100-000051FC	*Channel 1 status (buffer 0-63)
00005200-000052FC	*Channel 2 status (buffer 0-63)
00005300-000053FC	*Channel 3 status (buffer 0-63)
00005400-000054FC	*Channel 4 status (buffer 0-63)
00005500-000055FC	*Channel 5 status (buffer 0-63)
00005600-000056FC	*Channel 6 status (buffer 0-63)
00005700-000057FC	*Channel 7 status (buffer 0-63)
00005800-000058FC	*Channel 8 status (buffer 0-63)
00005900-000059FC	*Channel 9 status (buffer 0-63)
0001xxxx	*Output data FIFO (non-pipelined mode) [d32, BLT]
0002xxxx	*Output data FIFO (pipelined mode) [BLT,MBLT]

Register Definitions

VRB ID

Always 3 for a VRB module.

Application ID

This register displays the current selected VRB application. For the CDF SVX application, the value should be 1.

Date Code

The version of the VRB firmware for the application currently running is indicated by a 4 digit hexadecimal date code. The format is YMDD where;

Y = last digit of year

M = month (January = "1".....December = "C")

DD = day

Module Serial Number

Individual board serial number for this VRB.

Module Type

Contains a value of 1 if this is a 10 channel X 64 Kbyte board (normal CDF configuration), 0 if this is an 8 channel X 32 Kbyte board.

Configuration Switches

VRB configuration settings (value of switch S3-1,2,3,4).

Bit	**Function**
0	Bank 0 flash memory write enable (0 = write protected, 1 = write enabled)
1	Bank 1 flash memory write enable (0 = write protected, 1 = write enabled)
2	Diagnostic Port enable (0 = port disabled, 1 = port enabled)
3	Remote mode change enable (0 = mode change disabled, 1 = mode change enabled)
4-15	Not used

Readout Buffer Number

The VRB buffer number (6 bits) to use for readout of the current event. This information is normally supplied by the SRC through the system controller interface, but may also be written to this register through VME for test purposes. The VRB will return READOUT_BUSY when the Readout Buffer Number is received, but will not begin processing the readout until the Readout Buffer Number, Readout Pipeline Capacitor Number, and Readout Bunch Crossing Number are all received.

Readout Pipeline Capacitor Number

The pipeline capacitor number (8 bits) to use for consistency checking during readout of the current event. This information is normally supplied by the SRC through the system controller interface, but may also be written to this register through VME for test purposes.

Readout Bunch Crossing Number

The bunch crossing number (8 bits) to use for consistency checking during readout of the current event. This information is normally supplied by the SRC through the system controller interface, but may also be written to this register through VME for test purposes.

Scan Buffer Number

The VRB buffer number (6 bits) to use for output of the next event to the scan processor. This information is normally supplied by the SRC through the system controller interface, but may also be written to this register through VME for test purposes. The VRB will return SCAN_BUSY when the Scan Buffer Number is received, but will not begin processing the scan until both the Scan Buffer Number and Scan Event Number are received.

Scan Event Number

The event number (8 bits) to be inserted into the header field of the next scanned event. This information is normally supplied by the SRC through the system controller interface, but may also be written to this register through VME for test purposes.

Time Since L1

The time since L1 trigger (8 bits) to be inserted into the header field of the next scanned event. This information is normally supplied by the SRC through the system controller interface, but may also be written to this register through VME for test purposes.

Current Status

This register holds an unlatched copy of the 10 status signals driven on the VRB Fanout cable to the SRC. Error signals are normally asserted at the completion of readout and remain asserted until the next readout command is received.

The SRC receives the OR of the status signals from all VRBs in the system. The VRB generating a particular error signal can be identified by examining this register individually in each VRB.

Latched Status

This register holds a latched copy of the 10 status signals driven on the VRB Fanout cable to the SRC. If any of the status signals have been asserted since the last clear status command, the corresponding bit will be latched high in this register. Note that the READOUT_BUSY and SCAN_BUSY signals will be asserted in normal operation....these bits will always be latched high if the VRB has received at least one READOUT or SCAN command.

VRB Reset/Restart

The VRB can be programmed to recognize up to 256 reset/restart commands. These commands can be sent either through VME or by the SRC. The following commands are currently implemented;

0x00	reset input and output data FIFOs, VRB control and buffer logic (does not reset processor or reload FPGAs)
0x01	same as 0, but also resets VTM
0x02-0x7F	(reset) not defined
0x80	restart VRB with application 0
0x81	restart VRB with application 1
0x82	restart VRB with application 2
0x83	restart VRB with application 3
0x84	restart VRB with application 4
0x85	restart VRB with application 5
0x86	restart VRB with application 6
0x87	restart VRB with application 7

0x88	restart VRB in Default Mode
0x89-0xFF	(restart) not defined

The restart commands are provided to allow test software to switch between applications without removing the VRB and changing default switch settings. This overrides the settings of the VRB Application Select switches (S3-5,6,7,8) until the next restart. Note that restart commands are only recognized if the Remote Mode Change Disable switch (S3-4) is OFF.

Channel Enable

The VRB data channels may be individually enabled or disabled. Disabled channels have a byte count of zero in the event header and no data in the data record. Error signals (loss of sync, etc) are not processed if the channel is not enabled. The default startup setting for this register is all channels disabled (0). For all channels enabled, the correct value for this register would be 0x03FF.

Important: settings for the channel enable register do not take effect until the VRB is reset by writing to the VRB reset/restart register (address 78, reset command 0 or 1) or by sending an SRC reset command (command 14, value 0 or 1).

Bit number	Function
0	channel 0 enable
1	channel 1 enable
2	channel 2 enable
3	channel 3 enable
4	channel 4 enable
5	channel 5 enable
6	channel 6 enable
7	channel 7 enable
8	channel 8 enable
9	channel 9 enable
10-15	not used

Emulation Mode Enable

The VRB data channels may be individually placed in “real” or “emulated” data modes. In emulated data mode, a 256 byte incrementing pattern is generated in place of input data from the FIB

for the selected channel(s). The generated data pattern increments from 0x00 to 0x7F with the same count appearing in both halves of each 16 bit "SVX" word. The default startup setting for this register is all channels in "real" data mode (0).

Important: settings for the emulation mode enable register do not take effect until the VRB is reset by writing to the VRB reset/restart register (address 78, reset command 0 or 1) or by sending an SRC reset command (command 14, value 0 or 1).

Bit number	Function
0	channel 0 emulation mode
1	channel 1 emulation mode
2	channel 2 emulation mode
3	channel 3 emulation mode
4	channel 4 emulation mode
5	channel 5 emulation mode
6	channel 6 emulation mode
7	channel 7 emulation mode
8	channel 8 emulation mode
9	channel 9 emulation mode
10-15	not used

WYSIWYG Mode Enable

The VRB data channels may be individually placed in a mode where the input data stream is passed to the data buffer without processing. In this mode leading zero words, words containing "D0" fill characters, and data following the first "Cx" EOR word are written to the data buffer as received. The default startup setting for this register is all channels in standard data mode (0).

Important: settings for the WYSIWYG mode enable register do not take effect until the VRB is reset by writing to the VRB reset/restart register (address 78, reset command 0 or 1) or by sending an SRC reset command (command 14, value 0 or 1).

Bit number	Function
0	channel 0 WYSIWYG mode
1	channel 1 WYSIWYG mode
2	channel 2 WYSIWYG mode
3	channel 3 WYSIWYG mode
4	channel 4 WYSIWYG mode
5	channel 5 WYSIWYG mode

6	channel 6 WYSIWYG mode
7	channel 7 WYSIWYG mode
8	channel 8 WYSIWYG mode
9	channel 9 WYSIWYG mode
10-15	not used

Timeout Enable

When set, the data channel will timeout 20 usec after a readout command if no data has been received. The event will be terminated as if it were a normal zero-length record, but the timeout error signal will be asserted. The default startup setting for this register is timeouts disabled (0).

Important: settings for the timeoute enable register do not take effect until the VRB is reset by writing to the VRB reset/restart register (address 78, reset command 0 or 1) or by sending an SRC reset command (command 14, value 0 or 1).

Bit number	Function
0	channel 0 timeout enable
1	channel 1 timeout enable
2	channel 2 timeout enable
3	channel 3 timeout enable
4	channel 4 timeout enable
5	channel 5 timeout enable
6	channel 6 timeout enable
7	channel 7 timeout enable
8	channel 8 timeout enable
9	channel 9 timeout enable
10-15	not used

VRB Control

This register is used to control some global features of the VRB. Currently only one bit is assigned.

Bit number	Function
0	Control Port enable
1-15	not assigned

The control port enable bit allows the user to turn off the VRB control port (J5/6). This is normally only useful in test modes where there is no VRB Fanout module in the system and the VRB is being controlled through VME. When no VRB Fanout module is present, the control port message lines are not driven or terminated, so there is a possibility of spurious commands being received by the VRB if the port is enabled. This bit is ON by default.

Note: this bit controls VRB response to control messages on the J5/6 control port only. It does not enable/disable J5/6 status signals.

VTM Control

A command byte written to this register will be sent directly to the VTM for processing. The command format is as follows;

Bit number	Function
0	Read optical power level (A/D)
1	Read optical signal detect
2	Disable GLink clock
3	Enable GLink clock
4	Reset GLink
5-6	VTM Channel number (0-3)
7-15	not used

This register will be reset to zero as soon as the command is executed. Only ONE of the command bits (bits 0-4) may be set. If more than one bit is set, only the highest priority command will be executed (bit 0 = highest.....bit 4 = lowest priority).

Examples;

1) Read the optical power level on link 2

VTM Control Register = 0x0041

2) Reset link 3

VTM Control Register = 0x0070

Monitor Count

Returns the number of bytes in the monitor buffer for a monitor snapshot operation or the number of events histogrammed for a monitor histogram operation. In snapshot mode, the count is

updated after the data has been copied from the monitor FIFO to the monitor buffer (VME memory). In histogram mode, the count is updated continuously for each event histogrammed.

Monitor Channel

Used to set the VRB channel number (0-9) being monitored.

Monitor Control

This register determines the type of monitor operation performed. Defined operations are;

0	Idle	(no operation)
1	Reset	Clear the monitor buffer and reset the Monitor Count register. (the value in the Monitor Control register returns to 0 when the reset operation is complete)
2	Snapshot	Copy data from a single event to the monitor buffer. (the value in the Monitor Control register returns to 0 when the snapshot operation is complete)
3	Histogram	Generate a histogram of SVX channels (the histogram operation continues until the Monitor Control register is set back to 0 by the user.) The histogram table appears in the monitor buffer with the following order; Locations 0x1000- 0x11FC SVX chip 0, channels 0-127 Locations 0x1200- 0x12FC SVX chip 1, channels 0-127 ….. Locations 0x2E00-0x2FFC SVX chip 15, channels 0-127 (Note that SVX chips are ordered by their position in the data stream, not by the SVX chip ID. Channel 127 is not histogrammed if the SVX chip ID is 0.)

VTM Power

After transferring control to the CDF SVX application the VRB issues a sequence of commands to the VTM (via the VTM serial port) to read the Finisar optical receiver power. The measured power levels for links 0,1,2 and 3 are stored in separate registers (000000F0 through

000000FC, respectively). The low 8 bits of the register contain the power value read by the A/D converter. Bit 8 of the register is set if the Finisar optical receiver "signal detect" is set. If this is not set, the fiber optic cable may be disconnected or damaged. The power measurement and signal detect status are only checked once at startup or in response to a VTM control register write.....they are not continuous readings.

Channel Status Enables

Status flags for VRB channels can be individually enabled. Registers 00000140 through 00000164 contain the status enables for channels 0 through 9, respectively. The default value for these registers is 0xFF.

Bit number	Function
0	not used
1	not used
2	G-Link sync error enable
3	G-Link frame error enable
4	Invalid PLC or BC number error enable
5	Invalid channel number sequence error enable
6	not used
7	VRB input buffer overflow error enable
8-15	not used

Buffer Start Address (0-63)

Each VRB data channel has a total of 64 KBytes of buffer memory. This buffer memory is divided into one or more fixed size event buffers. The VRB control logic allows up to 64 buffers (the CDF SVX system is expected to use only 12 buffers). The starting address of each of the VRB event buffers may be individually programmed. Starting addresses for each buffer are the same for all ten VRB data channels. The current startup default setting for the CDF SVX configuration is 16 buffers of 4 KBytes each.

Buffer Number	**Register Address (hex)**
0	00000600
1	00000604

2	00000608
3	0000060C
4	00000610
5	00000614
6	00000618
7	0000061C
8	00000620
9	00000624
10	00000628
11	0000062C
...	...
63	000006FC

The following examples are for a system with 8 buffers;

Buffer Number	**Starting Address (hex)**	**Buffer Size**	**Comments**
0	0000	8 KBytes	8 equal sized buffers
1	2000	8 KBytes	
2	4000	8 KBytes	
3	6000	8 KBytes	
4	8000	8 KBytes	
5	A000	8 KBytes	
6	C000	8 KBytes	
7	E000	8 KBytes	

Buffer Number	Starting Address (hex)	Buffer Size	Comments
0	0000	4 KBytes	6 smaller buffers
1	1000	4 KBytes	
2	2000	4 KBytes	
3	3000	4 KBytes	
4	4000	4 KBytes	
5	5000	4 KBytes	
6	6000	20 KBytes	
7	B000	20 KBytes	and 2 larger buffers

The VRB logic places no restrictions on the starting address assigned to each buffer (buffers may overlap). The following example would allow operation in three different configurations without resetting the VRB. The configuration would be selected by using a different set of buffer numbers;

Configuration 1 buffer numbers 0-7

Configuration 2 buffer numbers 8-11

Configuration 3 buffer numbers 12-13

Buffer Number	Starting Address (hex)	Buffer Size	Comments
0	0000	8 KBytes	8 buffers of 8 KBytes each
1	2000	8 KBytes	
2	4000	8 KBytes	
3	6000	8 KBytes	
4	8000	8 KBytes	
5	A000	8 KBytes	
6	C000	8 KBytes	
7	E000	8 KBytes	
8	0000	16 KBytes	4 buffers of 16 KBytes each
9	4000	16 KBytes	

10	8000	16 KBytes	
11	C000	16 KBytes	
12	0000	32 KBytes	2 buffers of 32 KBytes each
13	8000	32 KBytes	

The current default startup buffer assignment is shown in the following table;

Buffer Number	**Starting Address (hex)**	**Buffer Size**	**Comments**
0	0000	4 KBytes	16 buffers of 4 KBytes each
1	1000	4 KBytes	
2	2000	4 KBytes	
3	3000	4 KBytes	
4	4000	4 KBytes	
5	5000	4 KBytes	
6	6000	4 KBytes	
7	7000	4 KBytes	
8	8000	4 KBytes	
9	9000	4 KBytes	
10	A000	4 KBytes	
11	B000	4 KBytes	
12	C000	4 KBytes	
13	D000	4 KBytes	
14	E000	4 KBytes	
15	F000	4 KBytes	

Buffer Size (0-63)

Each VRB data channel has a total of 64 KBytes of buffer memory. This buffer memory is divided into one or more fixed size event buffers. The VRB control logic allows up to 64 buffers (the CDF SVX system is expected to use only 12 buffers). The size of each of the VRB event buffers may be individually programmed and is the same for all ten VRB data channels. The current startup default setting for the CDF SVX configuration is 16 buffers of 4 KBytes each.

The buffer size parameter is used only to determine buffer overflow error. The VRB will truncate event data written to a buffer when the byte count exceeds the Buffer Size parameter for that buffer. Because buffer start addresses are not necessarily consecutive and may overlap, the buffer size is not automatically calculated from the buffer start addresses.

Buffer Number	**Register Address (hex)**
0	00000700
1	00000704
2	00000708
3	0000070C
4	00000710
5	00000714
6	00000718
7	0000071C
8	00000720
9	00000724
10	00000728
11	0000072C
...	...
63	000007FC

Output Data Interface

The output data FIFO can be read in two ways (using separate VME addresses). The "pipelined" mode is much faster, but has a minor access restriction.

To maximize transfer rate, the "pipelined" mode prefetches data from the output data FIFO and places it in pipeline registers. When the VRB receives a VME data strobe (DS), it can immediately transfer the contents of the last pipeline register to the VME data bus and return DTACK within 20 ns. Once the VRB begins a pipelined block transfer, it assumes that the Scan Processor will complete the block transfer before accessing other registers in that VRB. The block transfer may be interrupted by other VME activity, as long as it is not a random read/write operation to the specific VRB involved in the block transfer. If this happens, the last two data words which were prefetched from the output data FIFO before the block transfer was interrupted will be discarded.

The "non-pipelined" block transfer mode does not prefetch data and so does not have the same restriction, but is 2-3 times slower. D64 block transfers (MBLT) only operate in the pipelined mode.

Application 2 (CDF DAQ)

In the CDF DAQ application, the VRB acts as a buffer for data following a Level 2 trigger decision. Buffer management is provided by the VRB internal logic. A subrack may contain up to 16 VRB modules. Each VRB receives data from up to ten "Tracer" modules via a TAXI transition module.

The VRB startup application and some programming/diagnostic features are controlled by switch bank S3;

S3-1 and S3-2 are write protect switches for the VRB flash memory. These may normally be left OFF (not write protected) to enable in-system updates of the VRB firmware and software. Since the VRB software does not allow flash write cycles when in CDF DAQ mode, there is generally no danger of accidentally overwriting the flash memory.

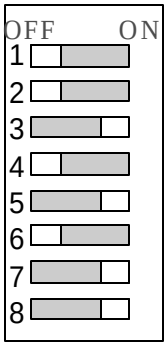
Switch S3-3 disables the RS-232 terminal diagnostic port. This port outputs information at VRB startup/reset and during some error conditions. If there is no terminal connected, this switch should be ON (disabled), since the VRB takes longer to recognize and execute a reset command when it is displaying diagnostic messages.

Switch S3-4 disables the remote mode change feature. When enabled (switch OFF), the VRB can be remotely rebooted into a mode other than CDF DAQ. This is necessary for remote reprogramming.

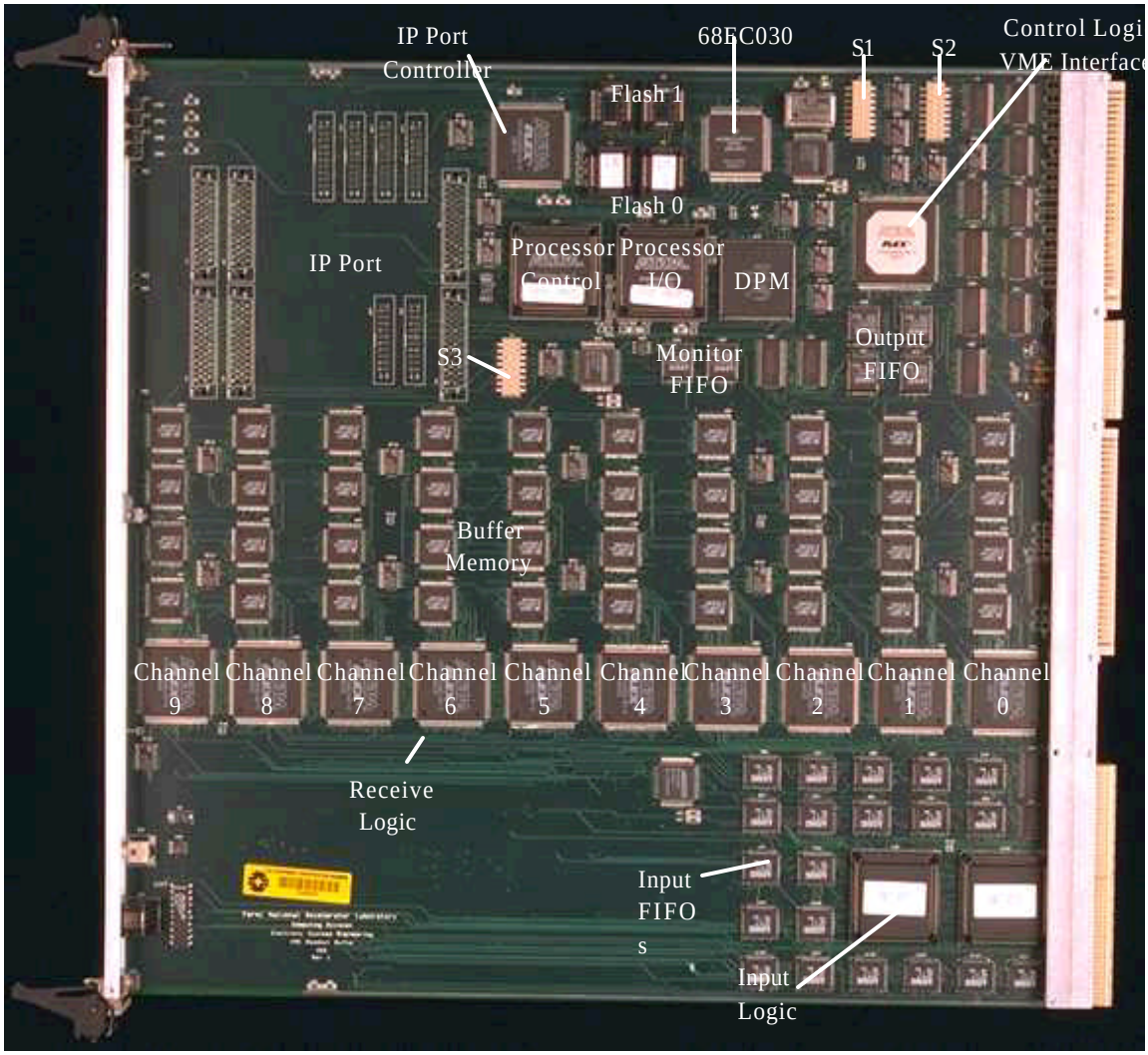
Switches S3-5 through S3-8 are used to select the VRB startup application. Application select switch settings for CDF DAQ mode are as follows;

	S3-8	**S3-7**	**S3-6**	**S3-5**
CDF DAQ	ON	ON	ON	OFF

Normal S3 settings for CDF DAQ mode



flash bank 0 write protect
flash bank 1 write protect
diagnostic port disable
remote mode change disable
application select 0
application select 1
application select 2
application select 3



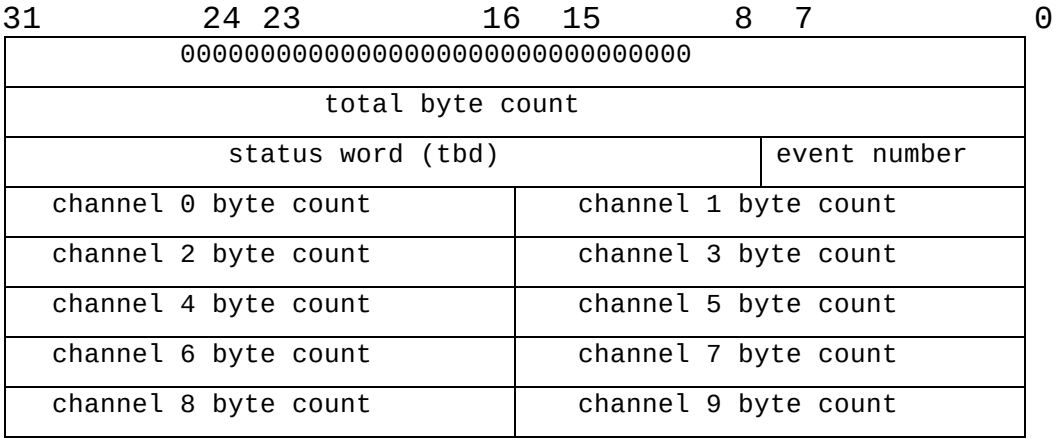
CDF DAQ VRB

For CDF DAQ applications, the VRB is used as a Level 3 input buffer and operates in FIFO mode. No data is discarded and no external controller is necessary. Event data is logically organized by longwords (4 bytes). The record is terminated by a stop bit (control bit in one of the four data bytes). Although the buffer memory operates as a FIFO, the VRB internally maintains read and write pointers for each event. There is an upper limit of 64 events resident in the buffer memory, based on the number of pointers available. Events are automatically transferred to the Output FIFO (space permitting) in the order received.

TAXI data links are used and the Receive Logic for each channel generates a BUSY signal when the buffer associated with that channel is "almost full" (256 Bytes short of the 64 KByte buffer capacity) or if all 64 event pointers for that buffer have been used. This BUSY signal is used to inhibit data transmission on the TAXI link.

Output Data Format

The following header information is added to each CDF DAQ event:



The event number is a VRB generated incrementing value (0-255). A trailer consisting of two zero words is added to the end of each event. If the output data FIFO is empty, subsequent read operations will return this last zero value. The VME Scan Processor can poll the output data FIFO until it returns a non-zero value. This will be the first word (total byte count) of the next scanned event. Scanned events can be queued in the output data FIFO and read at the maximum rate of the Scan Processor.

Scan Processing

The VRB is programmed to allow scan operations using a single VRB address (output data FIFO) and transfer mode (block transfer). A typical scan operation is a two step process;

1) read one 64 bit word from the output data FIFO (using VME D64 block transfer mode).....repeat this step until a non-zero value is read.....this value will be the total byte count for the event

2) use the total byte count to read the remainder of the event (using VME D64 block transfer mode)

Note that the value of the “total byte count” includes itself, therefore the correct block size to transfer in step 2 is “total byte count” minus eight.

The event header is 32 bytes long and the data from each channel is padded to an eight byte boundary. The “total byte count” includes all header, data and padding.

The byte count for each individual channel represents only the actual data for that channel (it does not include any padding bytes at the end of the channel’s data). To find the start of the next channel’s data, round the byte count for the previous channel to the next eight byte boundary.

Restart Operation

A VRB restart is generated under the following conditions;

1) power ON

2) front panel restart button

3) VME SYSRESET signal

4) expiration of the VRB processor watchdog timer (if enabled)

5) a VME remote restart command (if enabled)

"Restart" causes the VRB processor to reset which in turn reloads all FPGA programs for the selected application. The processor reinitializes shared memory parameters and clears the input and output data FIFOs. The restart process takes several seconds to download FPGA code. Completion of restart is indicated by the front panel Processor LED.

The VRB will also respond to various reset commands which can be transmitted through VME (VRB Reset/Restart register). These commands are only recognized if the processor and VME interface FPGA programs are functioning correctly.

VME Addressing

The CDF DAQ application uses A32 addressing with the following address modifiers;

0x09 or 0x0D (single word accesses)

0x0B or 0x0F (D32 block transfers)

0x08 or 0x0C (D64 block transfers)

The only VRB location that responds to block transfers is the output data FIFO.

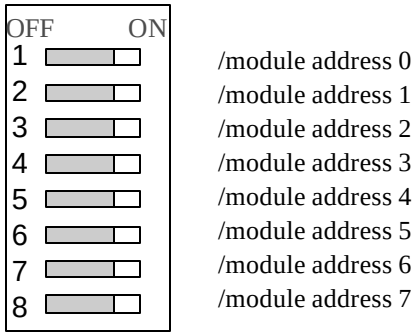
To conserve FPGA resources, D16 and D8 single word transfers are not currently implemented (as required by the VME standard).

The base address of the module is determined by the subrack geographical address (GA4:0).

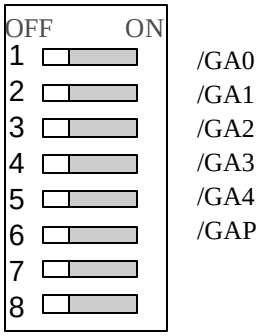
Address	**A32 Decoding**
A31	GA4
A30	GA3
A29	GA2
A28	GA1
A27	GA0

There are two VME address configuration switches. These are normally not used in the CDF DAQ mode, but must be set to the proper default positions. Switch bank S1 determines the eight bit module base address for systems not using geographical addressing (ON = 0, OFF = 1). For the CDF DAQ application, these switches are "don't care" but are normally set to the ON position. Switch bank S2 provides a means of setting the base address if geographical addressing is used, but is not supported by the subrack backplane (ON = 0, OFF = 1). These switches should all be set to OFF in the CDF DAQ application (when used with a standard VIPA subrack).

S1



S2



All registers are D32 unless otherwise indicated. Register addresses not listed are reserved. Registers that are written by the VRB and should be considered "read-only" for VME access are indicated by *.

VME Address (hex)	Function
00000000	*VRB ID
00000004	*Configuration
00000008	*Date code
0000000C	*Module serial number
00000010	*Module type
00000064	*TAXI violation count
00000070	*Current status
00000074	*Latched status
00000078	VRB reset/restart
000000E0	Channel enable
000000E4	Emulation mode enable
000001C0-000001E4	*Current read buffer number (channel 0-9)
00004000-000040FC	*Channel 0 byte counts (buffer 0-63)
00004100-000041FC	*Channel 1 byte counts (buffer 0-63)
00004200-000042FC	*Channel 2 byte counts (buffer 0-63)
00004300-000043FC	*Channel 3 byte counts (buffer 0-63)
00004400-000044FC	*Channel 4 byte counts (buffer 0-63)
00004500-000045FC	*Channel 5 byte counts (buffer 0-63)
00004600-000046FC	*Channel 6 byte counts (buffer 0-63)
00004700-000047FC	*Channel 7 byte counts (buffer 0-63)
00004800-000048FC	*Channel 8 byte counts (buffer 0-63)
00004900-000049FC	*Channel 9 byte counts (buffer 0-63)
00005000-000050FC	*Read completion flags (buffer 0-63)
0001xxxx	*Output data FIFO (non-pipelined mode) [d32, BLT]
0002xxxx	*Output data FIFO (pipelined mode) [BLT, MBLT]

Register Definitions

VRB ID

Always 3 for a VRB module.

Configuration Setting

This register displays the current selected VRB application. For the CDF DAQ application, the value should be 2.

Date Code

The version of the VRB firmware for the application currently running is indicated by a 4 digit hexadecimal date code. The format is YMDD where;

Y = last digit of year

M = month (January = "1".....December = "C")

DD = day

Module Serial Number

Individual board serial number for this VRB.

Module Type

Contains a value of 1 if this is a 10 channel X 64 Kbyte board (normal CDF configuration), 0 if this is an 8 channel X 32 Kbyte board.

TAXI Violation Count

A 16 bit count of the number of TAXI violations detected since module reset. This count represents the OR of all ten receivers.

Current Status

This register holds an unlatched copy of the current VRB status signals (TBD).

Latched Status

This register holds a latched copy of the VRB status signals (TBD). If any of the status signals have been asserted since the last clear status command, the corresponding bit will be latched high in this register.

VRB Reset/Restart

The VRB can be programmed to recognize up to 256 reset/restart commands. The following commands are currently implemented;

0x00	reset input and output data FIFOs, VRB control and buffer logic (does not reset processor or reload FPGAs)
0x01-0x7F	(reset) not defined
0x80	restart VRB with application 0
0x81	restart VRB with application 1
0x82	restart VRB with application 2
0x83	restart VRB with application 3
0x84	restart VRB with application 4
0x85	restart VRB with application 5
0x86	restart VRB with application 6
0x87	restart VRB with application 7
0x88	restart VRB in Default Mode
0x89-0xFF	(restart) not defined

The restart commands are provided to allow test software to switch between applications without removing the VRB and changing default switch settings. This overrides the settings of the VRB Application Select switches (S3-5,6,7) until the next restart. Note that restart commands are only recognized if the Remote Restart Disable switch (S3-4) is OFF.

Channel Enable

The VRB data channels may be individually enabled or disabled. Disabled channels have a byte count of zero in the event header and no data in the data record. Error signals (violation, etc) are not processed if the channel is not enabled. The default startup setting for this register is all channels disabled (0). For all channels enabled, the correct value for this register would be 0x3FF.

Important: settings for the channel enable register do not take effect until the VRB is reset by writing to the VRB reset/restart register (address 78, reset command 0 or 1).

Bit number	Function
0	channel 0 enable

1	channel 1 enable
2	channel 2 enable
3	channel 3 enable
4	channel 4 enable
5	channel 5 enable
6	channel 6 enable
7	channel 7 enable
8	channel 8 enable
9	channel 9 enable
10-15	not used

Emulation Mode Enable

The VRB data channels may be individually placed in "real" or "emulated" data modes. In emulated data mode, a 256 byte incrementing pattern is generated in place of input data from the FIB for the selected channel(s). The generated data pattern increments from 00 to FF with the same count appearing twice in each 16 bit word. The default startup setting for this register is all channels in "real" data mode (0).

Important: settings for the emulation mode enable register do not take effect until the VRB is reset by writing to the VRB reset/restart register (address 78, reset command 0 or 1).

Bit number	Function
0	channel 0 emulation mode
1	channel 1 emulation mode
2	channel 2 emulation mode
3	channel 3 emulation mode
4	channel 4 emulation mode
5	channel 5 emulation mode
6	channel 6 emulation mode
7	channel 7 emulation mode
8	channel 8 emulation mode
9	channel 9 emulation mode
10-15	not used

Output Data Interface

The output data FIFO can be read in two ways (using separate VME addresses). The "pipelined" mode is much faster, but has a minor access restriction.

To maximize transfer rate, the "pipelined" mode prefetches data from the output data FIFO and places it in pipeline registers. When the VRB receives a VME data strobe (DS), it can immediately transfer the contents of the last pipeline register to the VME data bus and return DTACK within 20 ns. Once the VRB begins a pipelined block transfer, it assumes that the Scan Processor will complete the block transfer before accessing other registers in that VRB. The block transfer may be interrupted by other VME activity, as long as it is not a random read/write operation to the specific VRB involved in the block transfer. If this happens, the last two data words which were prefetched from the output data FIFO before the block transfer was interrupted will be discarded.

The "non-pipelined" block transfer mode does not prefetch data and so does not have the same restriction, but is 2-3 times slower. D64 block transfers (MBLT) only operate in the pipelined mode.

Application 3 (CDF SVX Test)

This is a special mode used for high rate data link test. Data is checked by the VRB for packet checksum errors.

The VRB startup application and some programming/diagnostic features are controlled by switch bank S3;

S3-1 and S3-2 are write protect switches for the VRB flash memory. These may normally be left OFF (not write protected) to enable in-system updates of the VRB firmware and software. Since the VRB software does not allow flash write cycles when in CDF SVX Test mode, there is generally no danger of accidentally overwriting the flash memory.

Switch S3-3 disables the RS-232 terminal diagnostic port. This port outputs information at VRB startup/reset and during some error conditions. If there is no terminal connected, this switch should be ON (disabled), since the VRB takes longer to recognize and execute a reset command when it is displaying diagnostic messages.

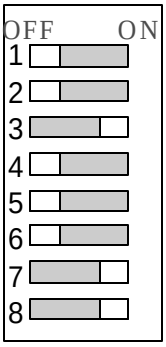
Switch S3-4 disables the remote mode change feature. When enabled (switch OFF), the VRB can be remotely rebooted into a mode other than CDF SVX Test. This is necessary for remote reprogramming, and for automated switching between CDF SVX and CDF SVX Test modes.

Switches S3-5 through S3-8 are used to select the VRB startup application. Application select switch settings for CDF SVX Test mode are as follows;

	S3-8	**S3-7**	**S3-6**	**S3-5**
CDF SVX Test	ON	ON	OFF	OFF

Normal S3 settings

for CDF SVX Test mode



flash bank 0 write protect
flash bank 1 write protect
diagnostic port disable
remote mode change disable
application select 0
application select 1
application select 2
application select 3

In CDF SVX Test mode, the VRB performs checksum calculations on the incoming data stream and uses the backplane status signals for checksum errors, but otherwise operates in the same way as the CDF SVX application. The checksum is calculated as follows;

1) add all data bytes received for an event up to and including the first "CX" EOR word, but not including any "D0" fill words.
2) subtract the value of the bunch crossing number sent by the SRC.
3) subtract N times the value of the pipeline capacitor number sent by the SRC, where N is a preset value for each channel (see register definitions).

The VRB drives the following status signals on the J5/6 backplane. These signals are active low (open collector TTL) on the backplane and are inverted by the VRB Fanout module.

- readout busy* (stat0): (same as SVX mode, held low if checksum error)
- scan busy* (stat1): (same as SVX mode)
- sync error* (stat2): (same as SVX mode)
- slot 13/15 checksum error* (stat3): Checksum error for VRBs in slots 13 and 15
- slot 12/16 checksum error* (stat4): Checksum error for VRBs in slots 12 and 16
- slot 11/17 checksum error* (stat5): Checksum error for VRBs in slots 11 and 17
- slot 10/18 checksum error* (stat6): Checksum error for VRBs in slots 10 and 18
- slot 9/19 checksum error* (stat7): Checksum error for VRBs in slots 9 and 19
- slot 8/20 checksum error* (stat8): Checksum error for VRBs in slots 8 and 20
- receive data timeout* (stat9): Indicates that a VRB is not receiving data...the VRB generating this error will also blink its front panel Controller status LED.

Following receipt of the last byte of data in an event, the checksum error status signals are driven high for approximately 400 ns if an error is detected.

All registers are the same as in the CDF SVX application with the following exceptions;

VME Address	**Function**
00000140-00000164	Checksum value (channel 0-9)
00000180-000001A4	Number of SVX chips (channel 0-9)

Checksum Value

Each register contains an 8 bit value to compare against the calculated input data stream checksum for the specified channel. The value may be different for each VRB channel. Only the low 8 bits of the calculated checksum are compared.

Number of SVX Chips

Each register contains a 4 bit value to indicate the number of SVX chips attached to the corresponding channel. It may be different for each VRB channel. When running the system with simulated data inputs (e.g., from a DEM module), the register is normally set to 0 (default value) since the pipeline capacitor number is a known fixed value and can be included in the expected checksum.

When running with real SVX chips, the pipeline capacitor number varies with each event and must be subtracted from the VRB calculated checksum to match the expected checksum value. This subtraction is performed once for each SVX chip connected to the VRB data channel.

Application 4 (D0 SVX Test)

The D0 SVX test application is identical to the CDF SVX Test application, but takes into account the reduction in number of channels and buffer memory.

The VRB startup application and some programming/diagnostic features are controlled by switch bank S3;

S3-1 and S3-2 are write protect switches for the VRB flash memory. These may normally be left OFF (not write protected) to enable in-system updates of the VRB firmware and software. Since the VRB software does not allow flash write cycles when in D0 SVX Test mode, there is generally no danger of accidentally overwriting the flash memory.

Switch S3-3 disables the RS-232 terminal diagnostic port. This port outputs information at VRB startup/reset and during some error conditions. If there is no terminal connected, this switch should be ON (disabled), since the VRB takes longer to recognize and execute a reset command when it is displaying diagnostic messages.

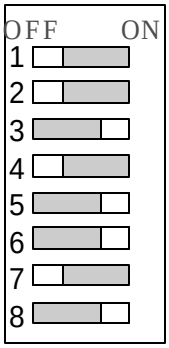
Switch S3-4 disables the remote mode change feature. When enabled (switch OFF), the VRB can be remotely rebooted into a mode other than D0 SVX Test. This is necessary for remote reprogramming.

Switches S3-5 through S3-8 are used to select the VRB startup application. Application select switch settings for D0 SVX Test mode are as follows;

	S3-8	**S3-7**	**S3-6**	**S3-5**
D0 SVX Test	ON	OFF	ON	ON

Normal S3 settings

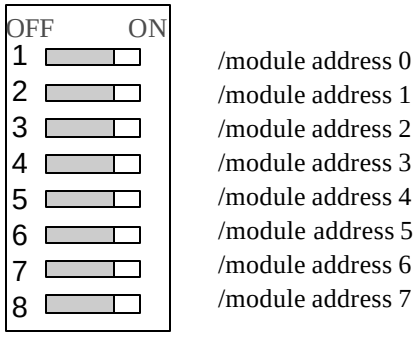
for D) SVX Test mode



flash bank 0 write protect
flash bank 1 write protect
diagnostic port disable
remote mode change disable
application select 0
application select 1
application select 2
application select 3

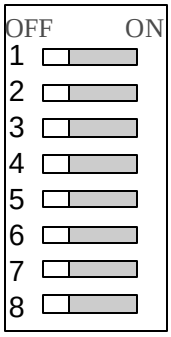
Two VME configuration switches are provided. S1 determines the eight bit module base address for systems not using geographical addressing (ON = 0, OFF = 1). S2 provides a means of setting the base address if geographical addressing is used, but is not supported by the subrack backplane (ON = 0, OFF = 1). Neither switch set is necessary in a VIPA standard subrack with geographical addressing (all S1 switches should be set to ON and all S2 switches should be set OFF in this case).

S1



/module address 0
/module address 1
/module address 2
/module address 3
/module address 4
/module address 5
/module address 6
/module address 7

S2



/GA0
/GA1
/GA2
/GA3
/GA4
/GAP

Application 5 (D0 Trigger)

The D0 Trigger application is identical to the D0 SVX application, but runs with the GLink VTM in 20 bit mode and uses GLink data lines D17 and D19 as end-of-record flags. The incoming data is actually 16 bits wide and is split between two VRB channels (odd bytes in one channel, even in the other). The data must then be reassembled after it has been read from the VRB.

The VRB startup application and some programming/diagnostic features are controlled by switch bank S3;

S3-1 and S3-2 are write protect switches for the VRB flash memory. These may normally be left OFF (not write protected) to enable in-system updates of the VRB firmware and software. Since the VRB software does not allow flash write cycles when in D0 Trigger mode, there is generally no danger of accidentally overwriting the flash memory.

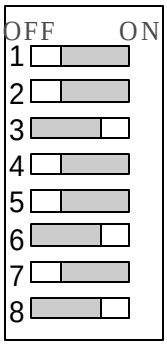
Switch S3-3 disables the RS-232 terminal diagnostic port. This port outputs information at VRB startup/reset and during some error conditions. If there is no terminal connected, this switch should be ON (disabled), since the VRB takes longer to recognize and execute a reset command when it is displaying diagnostic messages.

Switch S3-4 disables the remote mode change feature. When enabled (switch OFF), the VRB can be remotely rebooted into a mode other than D0 Trigger. This is necessary for remote reprogramming.

Switches S3-5 through S3-8 are used to select the VRB startup application. Application select switch settings for D0 Trigger mode are as follows;

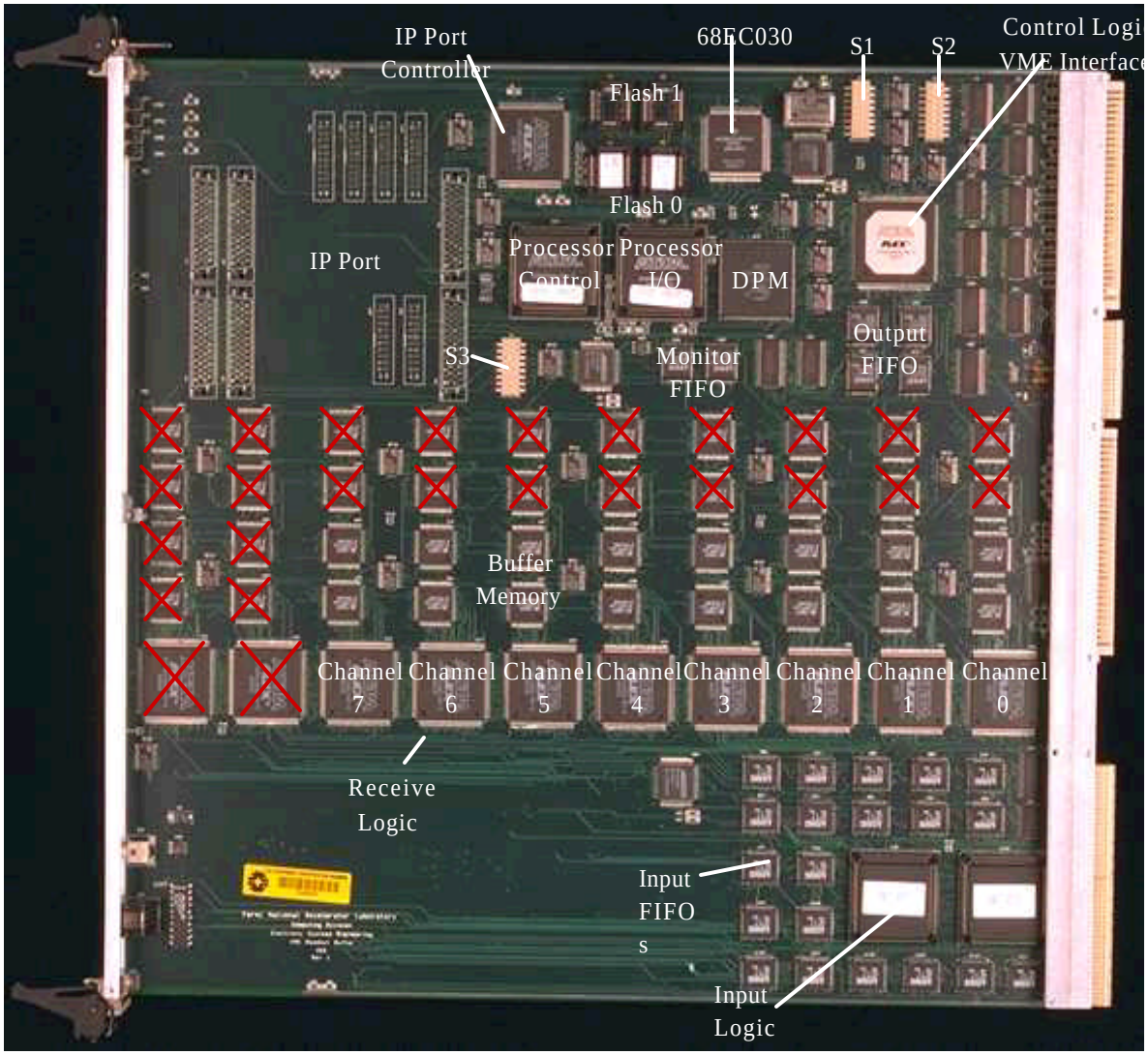
	S3-8	**S3-7**	**S3-6**	**S3-5**
D0 Trigger	ON	OFF	ON	OFF

Normal S3 settings for D0 Trigger mode



flash bank 0 write protect
flash bank 1 write protect
diagnostic port disable
remote mode change disable
application select 0
application select 1
application select 2
application select 3

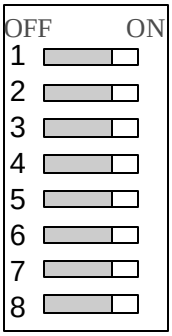
The chip locations in red are not populated on the D0 version of the VRB. Readout channel ordering is as shown.



D0 Trigger VRB

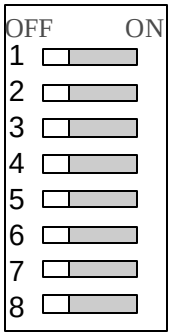
Two VME configuration switches are provided. S1 determines the eight bit module base address for systems not using geographical addressing (ON = 0, OFF = 1). S2 provides a means of setting the base address if geographical addressing is used, but is not supported by the subrack backplane (ON = 0, OFF = 1). Neither switch set is necessary in a VIPA standard subrack with geographical addressing (all S1 switches should be set to ON and all S2 switches should be set OFF in this case).

S1



/module address 0
/module address 1
/module address 2
/module address 3
/module address 4
/module address 5
/module address 6
/module address 7

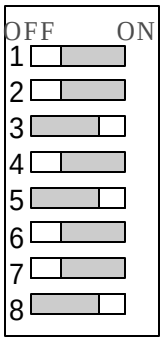
S2



/GA0
/GA1
/GA2
/GA3
/GA4
/GAP

Application 6 (unassigned)

S3



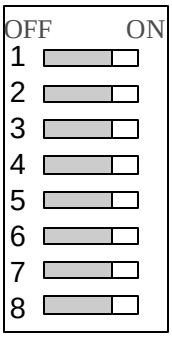
flash bank 0 write protect
flash bank 1 write protect
diagnostic port disable
remote mode change disable
application select 0
application select 1
application select 2
application select 3

Application select switch settings for Application 6 are as follows;

	S3-8	**S3-7**	**S3-6**	**S3-5**
Application 6	ON	OFF	OFF	ON

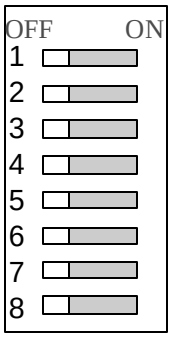
Two VME configuration switches are provided. S1 determines the eight bit module base address for systems not using geographical addressing (ON = 0, OFF = 1). S2 provides a means of setting the base address if geographical addressing is used, but is not supported by the subrack backplane (ON = 0, OFF = 1). Neither switch set is necessary in a VIPA standard subrack with geographical addressing (all S1 switches should be set to ON and all S2 switches should be set OFF in this case).

S1



/module address 0
/module address 1
/module address 2
/module address 3
/module address 4
/module address 5
/module address 6
/module address 7

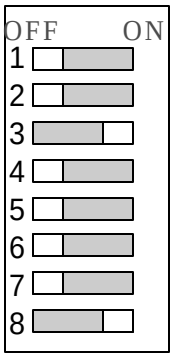
S2



/GA0
/GA1
/GA2
/GA3
/GA4
/GAP

Application 7 (unassigned)

S3



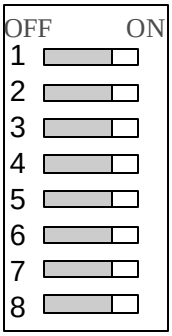
flash bank 0 write protect
flash bank 1 write protect
diagnostic port disable
remote mode change disable
application select 0
application select 1
application select 2
application select 3

Application select switch settings for Application 7 are as follows;

	S3-8	**S3-7**	**S3-6**	**S3-5**
Application 7	ON	OFF	OFF	OFF

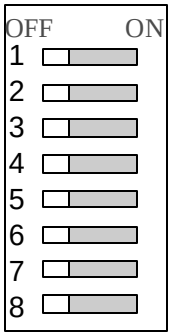
Two VME configuration switches are provided. S1 determines the eight bit module base address for systems not using geographical addressing (ON = 0, OFF = 1). S2 provides a means of setting the base address if geographical addressing is used, but is not supported by the subrack backplane (ON = 0, OFF = 1). Neither switch set is necessary in a VIPA standard subrack with geographical addressing (all S1 switches should be set to ON and all S2 switches should be set OFF in this case).

S1



/module address 0
/module address 1
/module address 2
/module address 3
/module address 4
/module address 5
/module address 6
/module address 7

S2



/GA0
/GA1
/GA2
/GA3
/GA4
/GAP

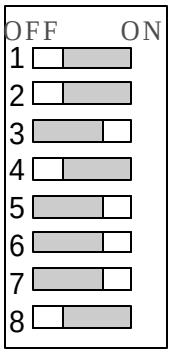
Default Mode

At startup the VRB will examine the setting of switch S3-8 (Application Select 3). If this switch is OFF, the VRB will be initialized with default code to allow generic VME access and diagnostics. **The VRB must be in Default mode to allow in-system reprogramming of the flash memory.**

Application select switch settings for the Default mode are as follows;

	S3-8	**S3-7**	**S3-6**	**S3-5**
Default	OFF	X	X	X

S3



flash bank 0 write protect
flash bank 1 write protect
diagnostic port disable
remote mode change disable
application select 0
application select 1
application select 2
application select 3

VME Addressing

The Default application uses A32 VME addressing with the following address modifiers;

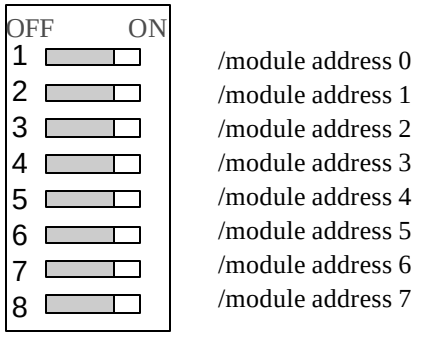
0x09 or 0x0D (single word accesses)

The base address of the module is determined by the subrack geographical address (GA4:0).

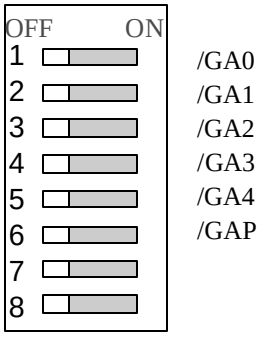
Address	**A32 Decoding**
A31	GA4
A30	GA3
A29	GA2
A28	GA1
A27	GA0

Two VME configuration switches are provided. S1 determines the eight bit module base address for systems not using geographical addressing (ON = 0, OFF = 1). S2 provides a means of setting the base address if geographical addressing is used, but is not supported by the subrack backplane (ON = 0, OFF = 1). Neither switch set is necessary in a VIPA standard subrack with geographical addressing (all S1 switches should be set to ON and all S2 switches should be set OFF in this case).

S1



S2



All registers are accessed using D32 single word transfers. These registers contain 16 bits of significant information in the low half of the 32 bit word with the high 16 bits returning zeros on reads and ignored on writes. Addresses not listed are reserved. Registers which are written by the VRB and should be considered "read-only" for VME access are indicated by *.

VME Address (hex)	**Function**
00000000	*VRB ID
00000004	*Configuration setting
00000008	*Date code
0000000C	*Module serial number
00000010	*Module type
00000078	VRB reset/restart
000000A0	Transfer command
000000A4	Transfer address low
000000A8	Transfer address high
00000800-0000BFC	Transfer data block

Register Definitions

VRB ID

Always 3 for a VRB module.

Configuration Setting

This register displays the current selected VRB application. For the Default application, the value should be 8.

Date Code

The version of the VRB firmware for the application currently running is indicated by a 4 digit hexadecimal date code. The format is YMDD where;

Y = last digit of year

M = month (January = “1”.....December = “C”)

DD = day

Module Serial Number

Individual board serial number for this VRB.

Module Type

Contains a value of 1 if this is a 10 channel X 64 Kbyte board, 0 if this is an 8 channel X 32 Kbyte board.

VRB Reset/Restart

The VRB can be programmed to recognize up to 256 reset/restart commands. These commands can be sent either through VME or by the System Controller. The following commands are currently implemented;

0x00-0x7F	(reset) not defined
0x80	restart VRB with application 0
0x81	restart VRB with application 1
0x82	restart VRB with application 2
0x83	restart VRB with application 3

0x84	restart VRB with application 4
0x85	restart VRB with application 5
0x86	restart VRB with application 6
0x87	restart VRB with application 7
0x88	restart VRB in Default Mode
0x89-0xFF	(restart) not defined

The restart commands are provided to allow test software to switch between applications without removing the VRB and changing default switch settings. This overrides the settings of the VRB Application Select switches (S3-5,6,7,8) until the next restart.

VRB Programming using VME

Programming the VRB through VME requires that the module be restarted in Default Mode. The reason for this is that the Default Mode firmware resides in the lower bank of flash memory where it will not be overwritten when the upper bank is reprogrammed. All other applications reside in the upper bank of flash memory.

Switching to Default mode is normally accomplished by writing a value of 0x88 to the VRB Reset/Restart register (the address of this register varies, depending on the current application). After sending the restart command, the programming software should wait approximately 5 seconds for the VRB to reinitialize.

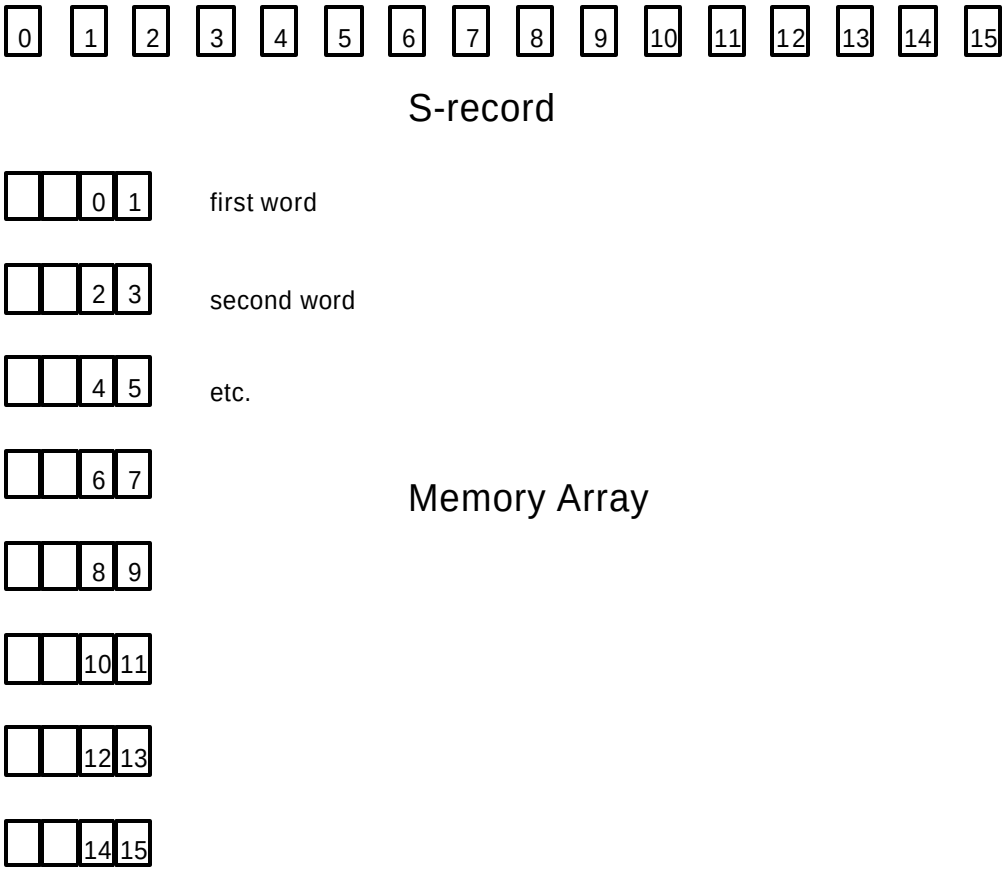
The programming software should then select and convert the appropriate S-record file. This file is usually named vrbXXXX.hex, where XXXX is the date code. This file contains all VRB applications and firmware, for both the upper and lower banks of flash memory.

Start by programming the upper bank of flash memory. Decode the S-record file into an integer array (the array should be 262,144 words, 32 bits each). An array should be used because the data in the S-record file is not necessarily in sequential order, but must be put in order before downloading to the flash memory. If you plan to verify after downloading, be sure to zero the array before decoding the S-record file. As you are decoding the S-record file, discard any lines with an address field BELOW 0x01000000 (these are part of the lower bank of flash memory). An sample line form a VRB S-record file is shown below, with spaces added between fields:

```
S3 15 01000000 4E 56 FF BC 48 E7 10 00 3D 7C 00 08 FF C6 3D 7C 0C
```

The S3 indicates a data record. The first record of the file will contain S0 and the last record will contain S7. The "15" is the (hex) number of remaining bytes in the record, including the address

field and the checksum. The next field is a 32 bit address indicating where the data should be put in flash memory. The next 16 bytes (in this example) are data and the last byte is a checksum. The data should be placed in the integer array as shown in the following picture (2 bytes in each word of the array);



Continue through the complete S-record file, placing each record in array locations according to the address field of the record. To translate between S-record address fields and array locations, subtract 0x01000000 from the S-record address field (for upper bank data records) and divide by 2 (to convert from byte addresses to array indices).

The data in this array must now be transferred to the upper bank of VRB flash memory in blocks of 256. Copy the first block to VME address 0x800 through 0xBFC using D32 single word transfers (note that even though you are using D32 transfers, only 16 bits of data are being transferred in each word….the high 16 bits of each word are unused).

After writing a block of data, set the starting address in flash memory where the data should be placed. Because the VRB registers are only 16 bits wide, you will have to set both an upper and lower transfer address register. Since we are starting with the first block of the upper bank of flash memory (address 0x01000000), set the upper transfer address register (VME location 0xA8) to 0x0100 and set the lower transfer address register (VME location 0xA4) to 0x0000.

Now write a command value of 1 to the transfer command register (VME location 0xA0). The VRB will handle the internal timing of the flash memory write which takes approximately 50 milliseconds. After the write is finished, the VRB will set the command register (VME location 0xA0) back to zero. You can poll this register to see when the write is finished.

The second block of the data array should be transferred to the same VRB buffer address (0x800 through 0xBFC), but this time the upper and lower flash memory transfer address registers should be set to 0x0100 and 0x0200 respectively (address 0x01000200). Increment the address registers by 0x200 for each block sent, until 1023 blocks are transferred. IMPORTANT: DO NOT transfer the final block of the array (starting at flash memory address 0x0107FE00) since this contains module specific information (serial number and configuration).

To verify that the upper bank of flash memory has been correctly programmed, set the upper and lower transfer address registers back to the start of the memory (0x0100 and 0x0000, respectively) and write a value of 2 to the transfer command register (VME location 0xA0). This will cause the data in the first block of flash memory to be copied to the VRB data buffer (0x800 through 0xBFC). The command register will be reset to zero when the transfer is complete. You can then read this block from the VRB using single word D32 transfers and compare it to the data in the array. Increment the address registers by 0x200 for each block as in the write case. Again skip the last block of the array since it will be different for each VRB.

Programming of the lower bank of flash memory is done in the same way. First though, the VRB processor must be told to jump to a location in the high bank of memory (since the flash memory cannot be programmed at the same time it is being used). Do this by sending a transfer command value of 5 (VME location 0xA0) before starting the download. This causes the processor to copy the flash memory driver to a location in high memory and then jump to that location.

You must reinitialize (zero) the data array in memory and make a second conversion pass through the S-record file. This time discard any S-records with address fields ABOVE 0x01000000 since these are for the upper bank of flash memory.

Transfer the data blocks to the VRB in the same way as was done for the upper bank of memory, using transfer commands of 1 to write a block and 2 to read a block The flash memory address for the lower bank starts at 0x00000000, so the transfer address registers should start at 0x0000 and 0x0000, and increment by 0x0200. The last block of the lower bank of memory is not used, so you may transfer the same number of blocks (1023) as in the upper bank.

When finished writing and verifying the lower bank of memory, the VRB must be restarted. This is because the original calling program has now been overwritten and the stack return addresses may no longer be valid. To restart the VRB, send a command value of 6 to the transfer command register. The VRB will restart using the application select switches, which should normally be configured for the application that was originally running.

Transfer command values of 3 and 4 are used to write and read the serial number/configuration block at the top of bank 1. This block is programmed when the module is first received and does not need to be reprogrammed unless memory chips are replaced.

Note 1: when programming the lower bank of flash memory, a copy of the driver is written to an unused block of memory in the upper bank (starting at address 0x0106FE00). If you try to verify the upper bank of memory after programming the lower bank, you will receive a verify error for this block. To avoid this, either don't verify or ignore verify errors for this block. Since all blocks beginning at addresses 0x010XFE00 are unused (or reserved), you may skip programming and verifying of these blocks. Address 0x010XFE00 corresponds to the last block in each group of 128 blocks.

Note 2: all VRBs contain the code for all applications, so there is a single data file which is updated regardless of which applications have changed. Refer to the VRB update list for information on changes. The code for each application starts at a fixed location in the upper bank of flash memory (application 0 starts at 0x01000000, application 1 starts at 0x01010000, etc.). It is therefore possible to reprogram just one application without disturbing others. There is also a separate date code contained in each application to identify those that have been updated.

However, to avoid too many variations of installed firmware, the current practice is to reprogram all applications (as well as the lower bank of flash memory) when any one changes. Users should examine the VRB update list to determine if their application is affected by newer versions of the VRB data file. For obvious reasons, we prefer that all VRBs be updated occasionally to the most recent version (even if there have been no major changes to your application) so that we do not have to maintain multiple versions.